EXHIBIT 4

2017-2544

In The
United States Court Of Appeals
For The Federal Circuit

PUREPREDICTIVE, INC.,

Plaintiff – Appellant,

v.

H2O.AI, INC.,

Defendant – Appellee.

ON APPEAL FROM THE UNITED STATES DISTRICT COURT FOR THE NORTHERN DISTRICT OF CALIFORNIA, CASE NO. 3:17-CV-03049-WHO

BRIEF OF APPELLANT

Perry S. Clegg
R. Jeremy Adamson
KUNZLER, PC
50 W. Broadway
Suite 1000
Salt Lake City, UT 84101
(801) 994-4646

Counsel for Appellant
PurePredictive, Inc.

*Gibson*MOORE APPELLATE SERVICES, LLC
206 East Cary Street ♦ P.O. Box 1460 (23218) ♦ Richmond, VA 23219
804-249-7770 ♦ www.gibsonmoore.net

CERTIFICATE OF INTEREST

2017-2544 *PurePredictive, Inc. v. H2O.AI, Inc.*

Counsel for Appellant PurePredictive, Inc. certifies the following:

1. The full name of the party represented by me is: **PurePredictive, Inc.**

2. The name of the real party in interest (if the party named in the caption is not the real party in interest) represented by me is: **None**

3. All parent corporations and publicly held companies that own 10 percent or more of the stock of the party or amicus curiae represented by me are: **None**

4. The names of all law firms and the partners or associates that appeared for the party or amicus curiae now represented by me in the trial court or agency or are expected to appear in this court (and who have not or will not enter an appearance in this case) are: **Robert Gempeler. Please note that Kunzler Law Group is now Kunzler, P.C.**

5. The title and number of any case known to counsel to be pending in this or any other court or agency that will directly affect or be directly affected by this court's decision in the pending appeal. **None**

Dated: November 3, 2017 Respectfully submitted,

By: /s/ Perry S. Clegg
Perry S. Clegg
R. Jeremy Adamson

Counsel for Appellant PurePredictive, Inc.

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s):

Cases:

Statutes:

Rules:

STATEMENT OF RELATED CASES

Pursuant to Federal Circuit Rule 47.5, counsel for Plaintiff-Appellant PurePredictive, Inc. states:

1. There are no, nor have there been, any other appeals in or from this same civil action or proceeding in the lower court before this or any other appellate court.

2. The patent-in-suit is not the subject of any *inter partes* review proceeding.

3. There are no, nor have there been, any case known to counsel to be pending in this or any other court that will directly affect or be directly affected by this court's decision in the pending appeal.

STATEMENT OF JURISDICTION

The district court had jurisdiction under 28 U.S.C. §§ 1331 and 1338. PurePredictive appealed from the final judgment entered on August 30, 2017. (Joint Appendix ("Appx") Appx1; Appx226-227). This Court has jurisdiction under 28 U.S.C. § 1295.

STATEMENT OF ISSUES

1. Whether the district court legally erred in holding that independent claims 1, 14, 17, and 23 were merely "directed to the abstract concept of the manipulation of mathematical functions and make use of computers only as tools, rather than provide a specific improvement on a computer-related technology."

2. Whether the district court legally erred by not imposing a "clear and convincing" standard of proof of invalidity under 35 U.S.C. § 282 when finding the patent claims invalid under 35 U.S.C. § 101.

3. Whether the district court legally erred by dismissing the Complaint without independently considering the validity of any of the dependent claims of the patent-in-suit.

STATEMENT OF CASE

This appeal involves U.S. Patent No. 8.880,446 ("the '446 Patent"). (Appx14-36) The patent relates to the application of machine learning to predictive analytics data processing technology. More specifically is relates to generating improved

predictive ensembles for processing predictive analytics data. The patent provides for predictive ensembles having greater confidence metrics, reduced noise, optimized overhead, and greater effectiveness and efficiency. The claimed predictive ensemble includes a structured environment that defines relationships between subsets of workload data and different subsets of learned functions in an array of combined learned functions. The claimed technology is necessarily rooted in computer technology and overcomes problems arising in the realm of computer networks.

PurePredictive filed suit on May 27, 2017, accusing H20.AI Inc.'s machine learning platform of infringing "one or more claims" of the '446 Patent". (Appx48:11, Appx14-36) H20.AI filed a motion to dismiss under Rule 12(b)(6), Fed. R. Civ. P., on June 14, 2017 relying solely on intrinsic evidence. (Appx122-135) No extrinsic evidence was proffered as proof of invalidity of the asserted claims. (*Id.*) The district court granted the motion to dismiss on August 29, 2017 without the benefit of a Markman hearing or any discovery related to claim construction. The district court accepted as true defendant's conclusory explanations regarding the alleged ineligibility of the claims under § 101. (Appx2-13). PurePredictive appeals from the district court's dismissal order holding the '446 Patent claims invalid under § 101.

Statement of The Facts

A. Background

PurePredictive is a technology-based service company that provides advanced analytics services using predictive ensembles constructed using artificial intelligence (e.g., machine learning). (Appx47:15-17; Appx24, 1:49-53, 1:59-62, 2:4-7, 2:16-19; Appx26, 6:25-29) The '446 Patent is vital for protecting PurePredictive's proprietary predictive-analytics technology, including its proprietary predictive analytics factory technology that uses machine-learning to generate unique and highly effective predictive ensembles. (Appx24, 1:49-52; Appx26:25-29, Exhibit A to Appellant-Plaintiff's Motion for Judicial Notice ("Mot. for Judicial Notice") 18:¶ 1, 74:¶ 8, 81:¶ 1, 83:¶ 6) PurePredictive's predictive ensembles are not mathematical formulations. They are structured environments that define relationships between classes or subsets of workload data and different combined learned functions within an arrangement of learned functions within the predictive ensemble.

B. PurePredictive's patent-in-suit

The invention claimed in the '446 Patent provides for an improved predictive ensemble that makes possible the processing of different subsets of data by different synthesized learned functions within the predictive ensemble. The claimed technology is necessarily rooted in computer technology and overcomes problems

arising in the realm of computer networks. As disclosed in the '446 Patent, a predictive analytics factory uses machine learning to produce these unique predictive ensembles. The predictive ensembles of the '446 Patent process workload data "to obtain a result, such as a classification, a confidence metric, an inferred function, a regression function, an answer, a prediction, a recognized pattern, a rule, a recommendation, or the like." (Appx27, 8:19-23; Appx32, 18:13-17)

But a predictive ensemble, as described in the '446 Patent, is not a math formula. FIG. 3 of the patent-in-suit, shown below, illustrates a representative predictive analytics module having a predictive ensemble as taught by the '446 Patent.

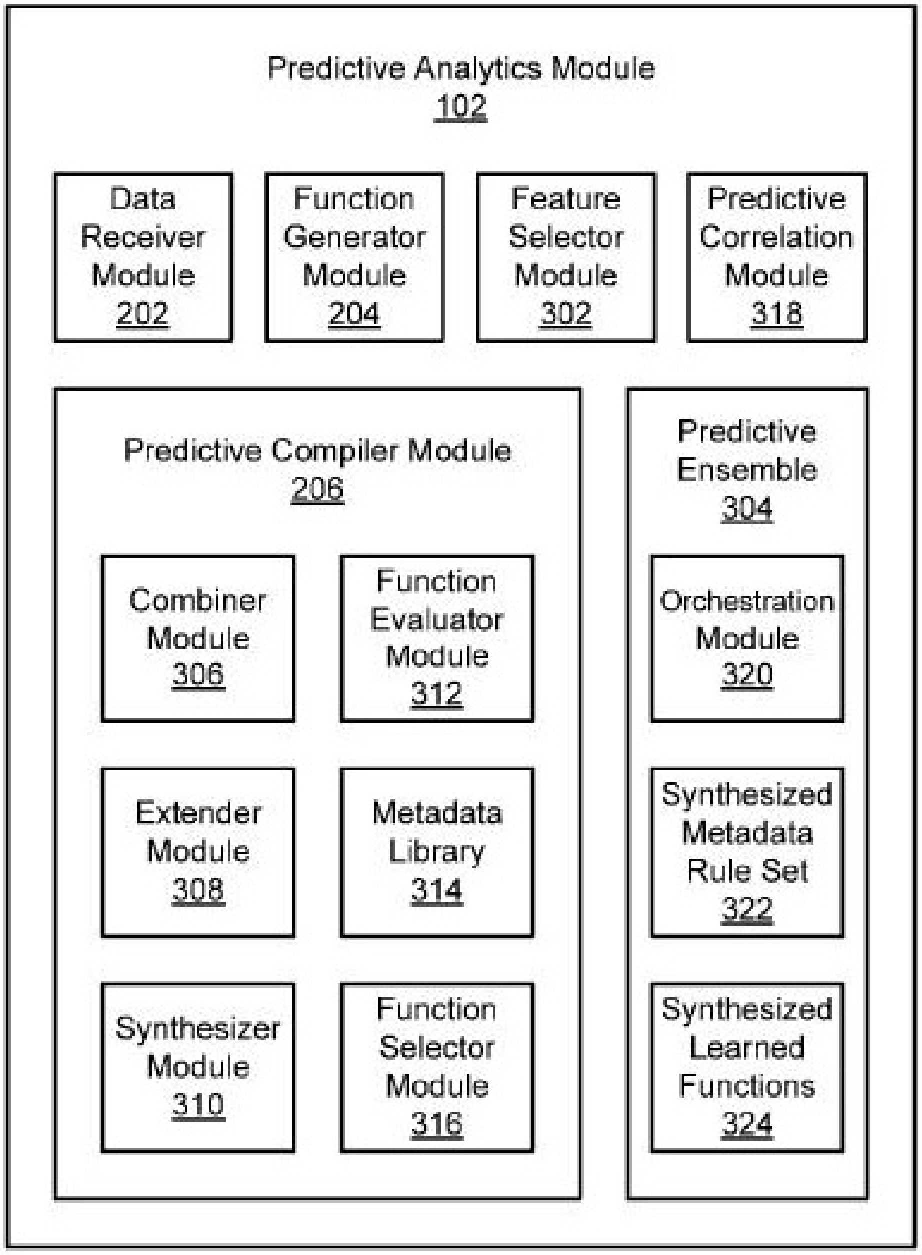


FIG. 3

Notably, the predictive ensemble is an assembly or "structure" comprised of (i) an orchestration module, (ii) a synthesized metadata rule set, and (iii) an arrangement of learned functions that together organize the workload and/or

workflow for processing of workload data performed by the array of learned functions. (Appx18; Appx19; Appx28, 10:57-59; Appx29, 11:41-44).

As described in the '446 Patent (col. 18, ll. 23-28):

> the synthesized metadata rule set 322 comprises a set of rules or conditions from the evaluation metadata of the metadata library 314 that indicate to the orchestration module 320 which features, instances, or the like should be directed to which synthesized learned function 324.

(Appx32, 18:23-28)

The synthesized metadata rule set is generated using a synthesizer module. As described in the '446 Patent (col. 18, ll. 33-39):

> The synthesizer module 310 may use that evaluation metadata to determine rules for the synthesized metadata rule set 322, indicating which features, which instances, or the like the orchestration module 320 the orchestration module 320 [sic] should direct through which learned functions, in which order, or the like.

(Appx32, 18:33-39)

Thus, the synthesized metadata rule set defines the direction of flow for the workload data and dictates the division of work load among learned functions. The orchestration module implements the metadata rules set and directs the flow of workload data for analytics processing to the appropriate learned functions depending on the class or subset of data.

As described in the '446 Patent (col. 18, ll. 11-22):

> The orchestration module 320 . . . is configured to direct workload data through the predictive ensemble 304 . . . [and] uses evaluation metadata

> from the function evaluator module 312 and/or the metadata library 314, such as the synthesized metadata rule set 322, to determine how to direct workload data through the synthesized learned functions 324 of the predictive ensemble 304.

(Appx32, 18:11-22)

FIG. 5 of the '446 Patent, below, illustrates representative arrangements of learned functions in a predictive ensemble as disclosed in the patent-in-suit.

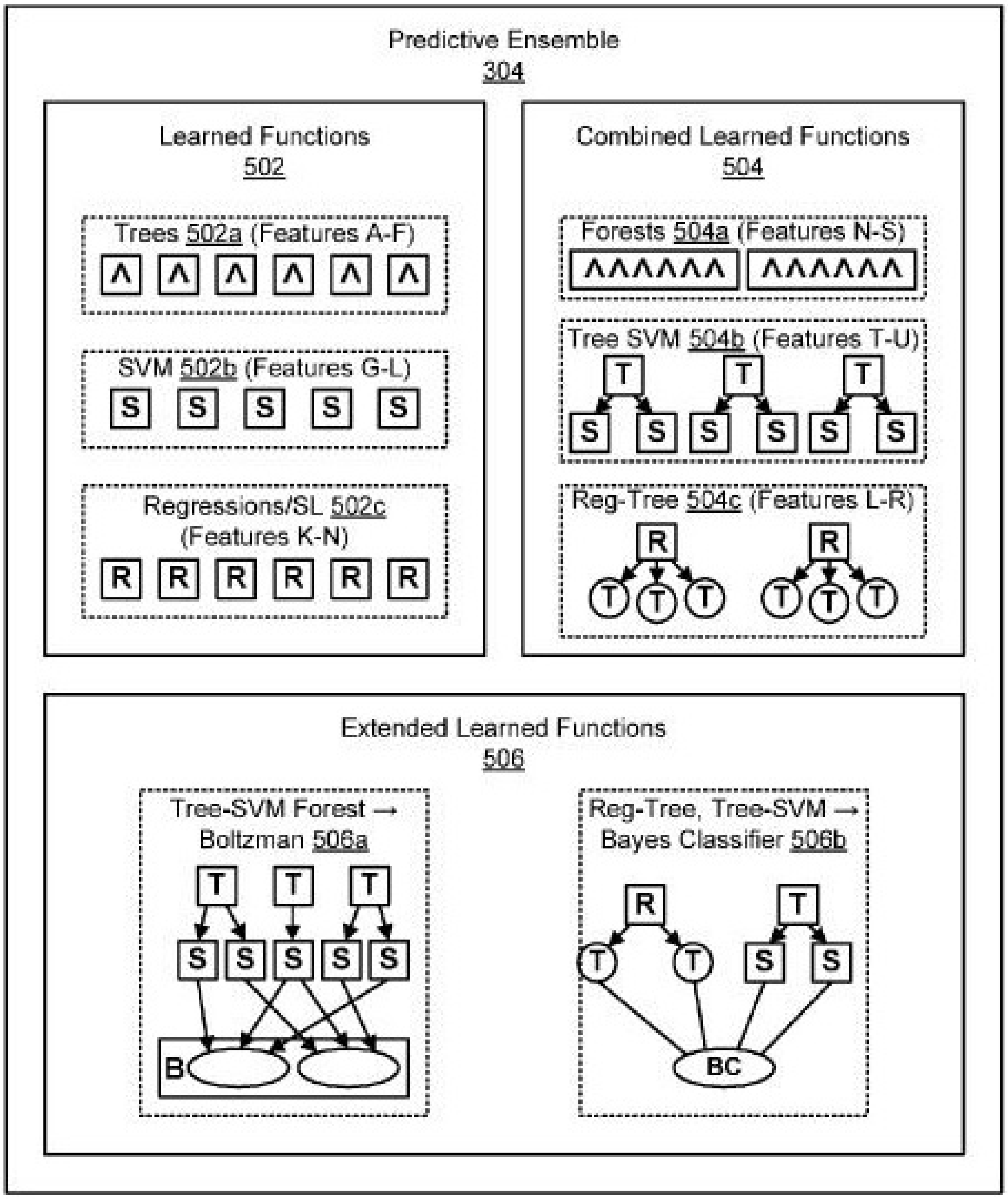


FIG. 5

As shown in FIG. 5 and described in the patent (col. 20, ll. 54-62), the set of learned functions may include:

> different classes [of learned functions] including a collection of decision trees 502a, configured to receive or process a subset A-F of the feature set of the predictive ensemble 304, a collection of support vector machines ("SVMs") 502b with certain kernels and with an input

> space configured with particular subsets of the feature set G-L, and a selected group of regression models 502c, here depicted as a suite of single layer ("SL") neural nets trained on certain feature sets K-N.

(Appx33, 20:54-62)

The learned functions of the predictive ensemble are not merely a collection of the fastest possible functions selected by the predictive analytics factory. Instead they are a structured arrangement of learned functions where different subsets of workload data may be assigned to different learned functions within the predictive ensemble. The data processing workload structure (e.g., which class or subset of data is directed to which learned function) is dictated by the synthesized metadata rules set and implemented by the orchestration module.

Likewise, the combined learned functions are not merely an assembly line of learned functions, such as a voting system. In analytics voting systems, each function is given the same data for processing. The results from each function are tallied to produce an overall predictive result. For example, in a group of 100 learned functions, each of the functions would be given the same data to process. If 55 of the functions predicted "yes" and 45 of the functions predicted "no", then the overall predictive "vote" would be "yes."

The data processing structure taught by the '446 Patent is an improvement over predictive voting systems. Where a predictive voting system is linear in nature, the '446 predictive ensemble is more of a three-dimensional arrangement. Indeed,

the distinction was specifically discussed during the ‘446 Patent prosecution. In an April 8, 2014 Office Action, the Patent Office rejected some of the claims as being unpatentable under 35 U.S.C. § 103(a) in view of US 2008/0162487 A1 by Richter (“*Richter*”). (Exhibit A to Mot. For Judicial Notice, 42:¶ 6) But *Richter* taught an ensemble having a voting system. In a June 11, 2014 After Final Amendment and Response to Office Action (pgs. 15-16), the ‘446 Patent applicant stated:

> The present amendments clarify that the synthesized rule set directs data such that different learned functions of the ensemble process different subsets of the data, such as different features/columns of data, different rows/instances of data, or both. For example, Fig. 5 of the Application depicts one embodiment of a predictive ensemble where different learned functions (e.g., learned functions 502, combined learned functions 504, and extended learned functions 506) process different subsets of data (e.g., different features/columns of data rows/instances, Features A-F, Features G-L, Features K-N, Features N-S, Features T-U, Features L-R). The ensemble methods mentioned in Fig. 13 of Richter, however, "voting," "average," "weighted average," and "hueristic rules" [sic.] do not distinguish between learned functions, with all learned functions processing all data. Richter does not teach any synthesized rules for directing data through multiple learned functions of an ensemble so that different learned functions process different subsets of the data.

(Exhibit A to Mot. For Judicial Notice, 75-76: ¶ 11)

The distinction between the ‘446 predictive ensemble and a voting system reflects a significant improvement in predictive analytics data processing technology. The division of workload data-processing labor by the predictive ensemble of the ‘446 Patent, contrasted with other predictive analytics modules that employ concurrent processing of the same data by each learned function, has a

number of advantages. Predictive analytics data will get processed quicker. Less power will be required to process the data. And classes or subsets of workload data will get processed by the learned functions most suited for the data. The method taught by the patent-in-suit for generating predictive ensembles makes these improved predictive ensembles and advantages possible. FIG. 4 of the ‘446 Patent, shown below, illustrates the relationship between the predictive ensemble and the claimed predictive analytics factory.

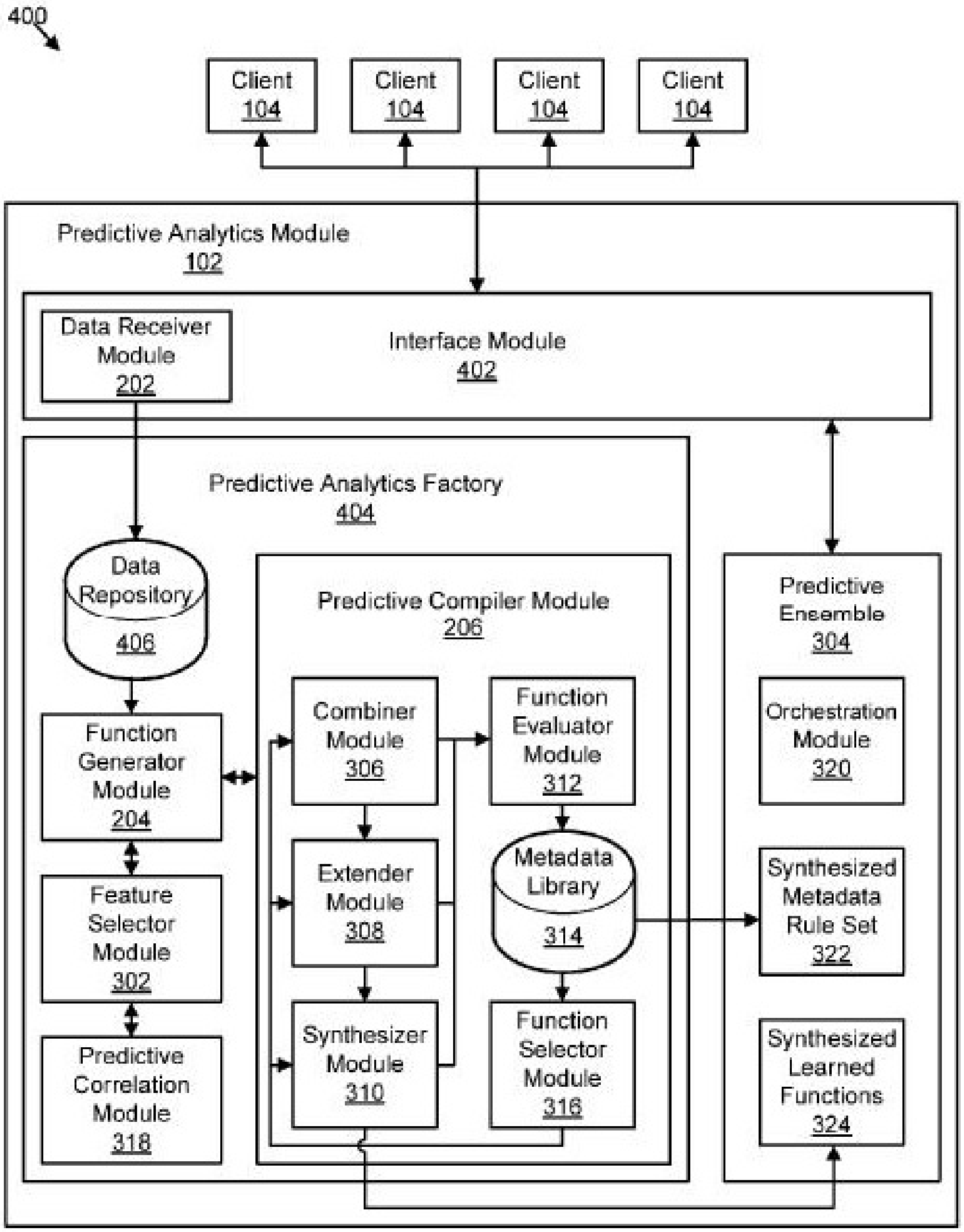


FIG. 4

The predictive analytics factory taught by the ‘446 Patent employs an intricate process for producing the claimed predictive ensembles. The method for forming the predictive ensemble does not merely mass produce learned functions and test to see

which ones are fastest. Even though numerous learned functions are generated at the early stages, significant changes are made during the process that improve the learned functions and the predictive ensemble when the learned functions are combined with the synthesized metadata rules set as reflected in FIG. 7 (below) of the '446 Patent.

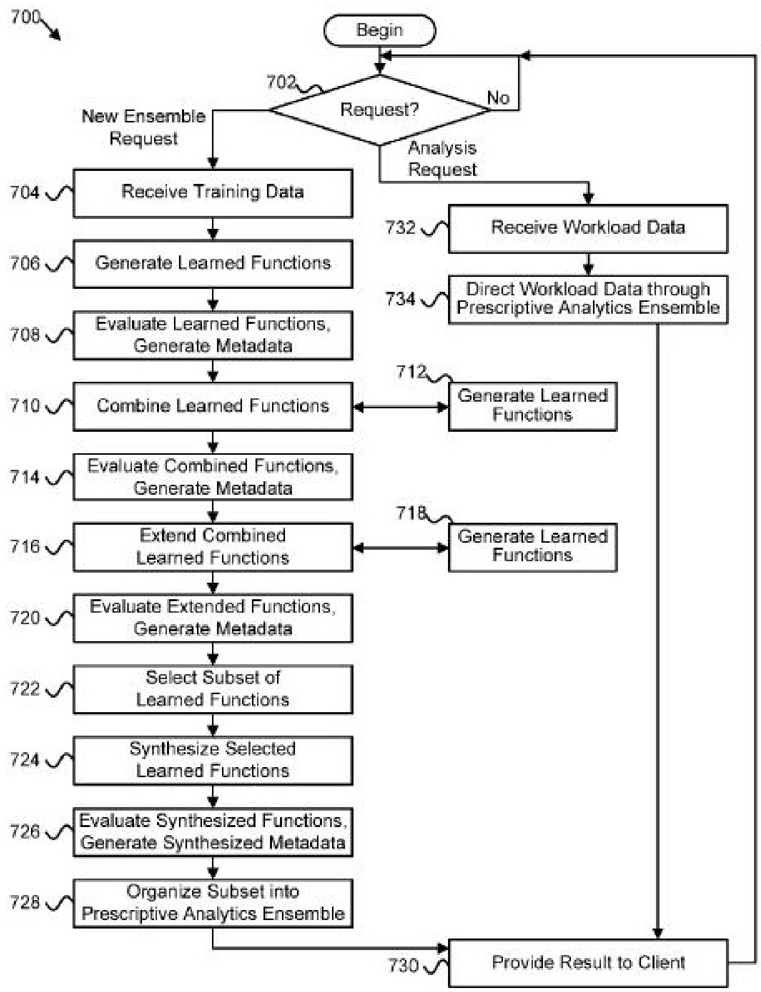


FIG. 7

After evaluation of pseudo-randomly generated learned functions for suitability and effectiveness, the selected learned functions are combined based on evaluation metadata, then the combined learned functions are evaluated again in

combination and additional evaluation metadata is generated. (Appx34, 21:37-39; Appx22) Combined learned functions are then extended. For example, as described in the '446 specification:

> the predictive compiler module 206 includes an extender module 308. The extender module 308, in certain embodiments, is configured to add one or more layers to a learned function. For example, the extender module 308 may extend a learned function or combined learned function by adding a probabilistic model layer, such as a Bayesian belief network layer, a Bayes classifier layer, a Boltzmann layer, or the like.

(Appx31, 15:1-8)

> Component class extended learned functions 506, extended by the extender module 308 or the like, include a set of extended functions such as a forest of trees 506a with tree decisions at the roots and various margin classifiers along the branches, which have been extended with a layer of Boltzmann type Bayesian probabilistic classifiers. Extended learned function 506b includes a tree with various regression decisions at the roots, a combination of standard tree 504b and regression decision tree 504c and the branches are extended by a Bayes classifier layer trained with a particular training set exclusive of those used to train the nodes.

(Appx34, 21:6-16)

Subsets of combined and extended learned functions are then configured to process or predict against different features (e.g., "combined and extended function 806 configured for features A-F or in the other case a different, parallel combined function 808 configured to predict against a feature set G-M" (Appx34, 22:30-31)).

The combined and extended learned functions are then organized into the predictive ensemble with metadata and a synthesized metadata rule set to define the

relationship among different combined learned function and the features and/or data subsets to be directed to the different learned functions within the predictive ensemble. As described in the ‘446 specification:

> the predictive compiler module 206 includes a synthesizer module 310. The synthesizer module 310 . . . is configured to organize a subset of learned functions into the predictive ensemble 304, as synthesized learned functions 324. In a further embodiment, the synthesizer module 310 includes evaluation metadata from the metadata library 314 of the function evaluator module 312 in the predictive ensemble 304 as a synthesized metadata rule set 322, so that the predictive ensemble 304 includes synthesized learned functions 324 and evaluation metadata, the synthesized metadata rule set 322, for the synthesized learned functions 324.
>
> The learned functions that the synthesizer module 310 synthesizes or organizes into the synthesized learned functions 324 of the predictive ensemble 304, may include . . . combined learned functions from the combiner module 306, extended learned functions from the extender module 308, combined extended learned functions, or the like.

(Appx31, 15:46-64).

The method for producing predictive ensembles described in the ‘446 specification provides improved predictive ensembles. By way of few examples, the method of the ’446 Patent provides for: (i) reduction of noise within the predictive ensemble; (ii) optimization of predictive ensemble overhead (e.g., size (number of functions) and complexity); and (iii) increased or optimizes confidence metrics. (Appx29, 12:55-59; Appx29, 11:45-60; Appx30, 13:19-27; Appx30, 13:35-39; Appx30, 14:9-25). These improvements to the ‘446 predictive ensembles are described in the ‘446 specification.

For example, the ‘446 specification describes how the invention reduces noise from predictive ensembles:

> [O]nce the feature selector module 302 determines that a feature is merely introducing noise, the predictive compiler module 206 excludes the feature from future iterations, and from the predictive ensemble 304.

(Appx29, 12:55-59)

The ‘446 specification also describes how the invention optimizes the overhead of predictive ensembles:

> the feature selector module 302 determines which features of initialization data to use in the predictive ensemble 304, and in the associated learned functions, and/or which features of the initialization data to exclude from the predictive ensemble 304, and from the associated learned functions. . . .
> Certain features may be more predictive than others, and the more features that the predictive compiler module 206 processes and includes in the generated predictive ensemble 304, the more processing overhead used by the predictive compiler module 206, and the more complex the generated predictive ensemble 304 becomes.

(Appx29, 11:45-60)

The ‘446 specification also describes how the invention improves or optimizes confidence metrics as follows:

> the predictive correlation module 318 determines one or more features, instances of features, or the like that correlate with higher confidence metrics (e.g., that are most effective in predicting results with high confidence). The predictive correlation module 318 may cooperate with, be integrated with, or otherwise work in concert with the feature selector module 302 to determine one or more features, instances of features, or the like that correlate with higher confidence metrics.

(Appx30, 13:19-27)

> The predictive correlation module 318 . . . is configured to harvest metadata regarding which features correlate to higher confidence metrics, to determine which feature was predictive of which outcome or result, or the like.

(Appx30, 13:35-39)

> In determining features that are predictive, or that have a greatest contribution to a predicted result or confidence metric, the predictive correlation module 318 may balance a frequency of the contribution of a feature and/or an impact of the contribution of the feature. For example, a certain feature or set of features may contribute to the predicted result or confidence metric frequently, for each instance or the like, but have a low impact. Another feature or set of features may contribute relatively infrequently, but has a very high impact on the predicted result or confidence metric (e.g. provides at or near 100% confidence or the like). While the predictive correlation module 318 is described herein as determining features that are predictive or that have a greatest contribution, in other embodiments, the predictive correlation module 318 may determine one or more specific instances of a feature that are predictive, have a greatest contribution to a predicted result or confidence metric, or the like.

(Appx30, 14:9-25)

Accordingly, there are a number of advantages to the method of producing predictive ensembles disclosed by the ‘446 Patent and improvements to the predictive ensembles because of the claimed methods.

Finally, it should also be noted that features claimed in the ‘446 Patent, such as the predictive analytics factory, the predictive ensemble, and other features are necessarily implemented with computer hardware and software. For example, the

predictive ensemble's metadata rule set is implemented using an orchestration module. As explained in the '446 Patent (col. 3, ll. 35-47):

> a module may be implemented as a hardware circuit . . . . A module may also be implemented in programmable hardware devices . . . .
>
> Modules may also be implemented in software for execution by various types of processors. . . . .

(Appx25, 3:35-47)

C. The claims at issue

The claims of '446 patent being asserted are consistent with the inventions described in the specification:

> 1. An apparatus for a predictive analytics factory, the apparatus comprising:
>
> a receiver module configured to receive training data for forming a predictive ensemble customized for the training data;
>
> a function generator module configured to pseudo-randomly generate a plurality of learned functions based on the training data without prior know ledge regarding suitability of the generated learned functions for the training data;
>
> a function evaluator module configured to perform an evaluation of the plurality of learned functions using test data and to maintain evaluation metadata for the plurality of learned functions, the evaluation metadata comprising one or more of an indicator of a training data set used to generate a learned function and an indicator of one or more decisions made by a learned function during the evaluation; and
>
> a predictive compiler module configured to form the predictive ensemble, the predictive ensemble comprising a subset of multiple learned functions from the plurality of learned functions, the multiple

learned functions selected and combined based on the evaluation metadata for the plurality of learned functions, the predictive ensemble comprising a rule set synthesized from the evaluation metadata to direct data through the multiple learned functions such that different learned functions of the ensemble process different subsets of the data based on the evaluation metadata.

(Appx34, 22:56 – Appx35, 23:18)

14. A method for a predictive analysis factory, the method comprising:

pseudo-randomly generating a plurality of learned functions based on training data without prior knowledge regarding suitability of the generated learned functions for the training data, the training data received for forming a predictive ensemble customized for the training data;

evaluating the plurality of learned functions using test data to generate evaluation metadata indicating an effectiveness of different learned functions at making predictions based on different subsets of the test data; and

forming the predictive ensemble comprising a subset of multiple learned functions from the plurality of learned functions, the subset of multiple learned functions selected and combined based on the evaluation metadata, the predictive ensemble comprising a rule set synthesized from the evaluation metadata to direct different subsets of the workload data through different learned functions of the multiple learned functions based on the evaluation metadata.

(Appx35, 24:21-41)

17. A computer program product comprising a non-transitory computer readable storage medium storing computer usable program code executable to perform operations for a predictive analysis factory, the operations comprising:

pseudo-randomly determining a plurality of learned functions using training data without prior knowledge regarding suitability of the determined learned functions for the training data, the training data

comprising a plurality of features, the training data received for forming a predictive ensemble customized for the training data;

selecting a subset of the features of the training data based on evaluation metadata generated for the plurality of learned functions, the evaluation metadata comprising an effectiveness metric for a learned function; and

forming the predictive ensemble, the predictive ensemble comprising at least two learned functions from the plurality of learned functions, the at least two learned functions using the selected subset of features, the at least two learned functions selected and combined based on the evaluation metadata, the predictive ensemble comprising a rule set synthesized from the evaluation metadata to direct data through the at least two learned functions so that different learned functions process different features of the selected subset of features.

(Appx35, 24:54 – Appx36, 25:10)

23. A predictive analytics ensemble comprising:

multiple learned functions synthesized from a larger plurality of learned functions, the multiple learned functions selected and combined based on evaluation metadata for an evaluation of the larger plurality of learned functions, wherein the larger plurality of learned functions are generated pseudo-randomly from training data without prior knowledge of a suitability of the larger plurality of learned functions for the training data;

a metadata rule set synthesized from the evaluation metadata for the plurality of learned functions for directing data through different learned functions of the multiple learned functions to produce a result; and

an orchestration module configured to direct the data through the different learned functions of the multiple learned functions based on the synthesized metadata rule set to produce the result.

(Appx36, 26:4-19)

PurePredictive alleged that defendant "infringes one or more claims" of the '446 Patent. PurePredictive's infringement action was intended to assert dependent as well as independent claims. For example, defendant is believed to infringe dependent claim 12.

(Appx35, 24:7-13)

D. The district court's dismissal order

The district court's dismissal order ("the Order") held that claims 1, 14, 17, and 23 were invalid under § 101 because they allegedly are "directed to the abstract concept of the manipulation of mathematical functions and make use of computers only as tools, rather than provide a specific improvement on a computer-related technology." (Appx2:17-19) The district court's decision invalidating the '446 Patent claims was made without the benefit of any discovery regarding claim construction.

The district court used claim 14 as a "representative claim" and characterized it as follows:

> Put very simply, the method performs predictive analytics in three steps. First, it receives data and generates "learned functions," or, for example, regressions from that data. . . . Second, it evaluates the effectiveness of those learned functions at making accurate predictions based on the test data. Finally, it selects the most effective learned functions and creates a rule set for additional data input. These three steps comprise the predictive analytics factory's method.

(Appx3:15-20)

The district court also characterized the claims as "directed to the patent-ineligible abstract concept of testing and refining mathematical algorithms" (Appx11:3-4) and merely "recit[ing] the functional steps for collecting, analyzing, and refining data through mathematical algorithms." (Appx13:3-4). The district court relied on these broad characterizations for its § 101 analysis without referring to or discussing the actual language of the claim elements, either individually or in combination. (Appx13:8-9)

The dismissal order also never references or discusses the "clear and convincing" evidence standard of proof the defendant bears to establish invalidity or the presumption of validity that attends a patent. Likewise, no dependent claims were ever evaluated or addressed by the Order. Rather, the Order simply states that "[a]mendment would be futile in light of the analysis above . . . ." (Appx13:12)

SUMMARY OF THE ARGUMENT

I. The Claims Cover Patent Eligible Subject Matter

The district court erred by granting the motion to dismiss holding that claims 1, 14, 17, and 23 are invalid under § 101. To the contrary, the claims are directed to an improvement in the technology of predictive analytics data processing. Each of the asserted claims includes a predictive ensemble that provides a structured environment for processing analytics data. Claim 1, for example, includes in relevant part:

> the predictive ensemble comprising a subset of multiple learned functions . . . selected and combined based on the evaluation metadata .

> . . , [and] a rule set synthesized from the evaluation metadata to direct data through the multiple learned functions such that different learned functions of the ensemble process different subsets of the data based on the evaluation metadata.

(Appx35, 23:8-18) (emphasis added)

The claimed predictive ensemble provides a structured environment that defines how various elements of information (e.g., subsets of data) are related to various combined learned functions. It is not a mathematical formula. Rather, metadata is used to define relationships between different subsets of workload data and different combined synthesized learned functions. A metadata rules set dictates which subsets of data get processed by which of the different subsets of learned functions within the predictive ensemble.

The district court described the '446 Patent's claims at such a high level of abstraction and untethered from the claim language that it ensured the claims would be patent ineligible. The district court's characterization of the claims removed core features of the claims, including the predictive ensemble. The district court also erred by finding the '446 Patent claims were invalid without applying a "clear and convincing" evidence standard or giving deference to the patent's presumption of validity. Likewise, the district court erred by not drawing all reasonable inferences from the '446 Patent intrinsic evidence alleged in the Complaint in favor of plaintiff, PurePredictive.

Thus, the order of dismissal should be reversed.

A. The district court oversimplified the claims without accounting for their core features

1. The district court oversimplified the claims in its § 101 analysis by finding they were directed to "the patent-ineligible abstract concept of testing and refining mathematical algorithms" (Appx11:3-4), or merely "recite the functional steps for collecting, analyzing, and refining data through mathematical algorithms." (Appx13:3-4)

2. The claims-at-issue are not directed to mathematical algorithms. Rather independent claims 1, 14, 17, and 23 are directed to a predictive analytics factory *for forming a predictive ensemble*, a method *for forming a predictive ensemble*, a computer program product *for forming a predictive ensemble*, and a *predictive ensemble* respectively. Notably, each of the claims includes a predictive ensemble consistent with the '446 specification.

3. A predictive ensemble is not a mathematical formula. Rather, as discussed above, it is a structured arrangement of (i) combined, synthesized learned functions, (ii) a metadata rules set, and (iii) means for implementation of the metadata rules, such as an orchestration module. Thus metadata is used to define relationships within the predictive ensemble so that an orchestration module can direct different classes or subsets of data to different combined learned functions for processing.

4. As a whole the claims-at-issue are not directed to excluded subject matter such as a common business method or a fundamental practice long prevalent that uses a general computer.

5. But the district court never meaningfully discussed the actual claim language in any of its § 101 analysis. Instead, in each instance the district court referenced to its overbroad abstraction, untethered from the actual language of the claims, and without consideration of the predictive ensemble. *CLS Bank Int'l v. Alice Corp. Pty. Ltd.*, 717 F.3d 1269, 1298 (Fed. Cir. 2013) *aff'd* 134 S. Ct. 2347 (2014)). Thus, the district court's § 101 analysis does not account for core features of the claims.

6. The claimed predictive ensemble feature is comparable to *Enfish's* logical model for a computer database that defined how the various elements of information in the database are related to each other (e.g., the "self-referential" database model), which was held as patent eligible by this Court. *Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327, 1330-33, 1346 (Fed. Cir. 2016).

7. Also, the claimed inventions of '446 Patent (*e.g.*, claims 1, 14, and 17) transform pseudo-randomly generated learned functions and metadata into the structured data processing environment of the '446 predictive ensemble. Thus, they are at least patent eligible under § 101 pursuant to "the machine-or-transformation

test." *In re Bilski*, 545 F.3d 943, 956, 963 (Fed. Cir. 2008) (en banc), *aff'd sub nom. Bilski v. Kappos*, 561 U.S. 593 (2010).

8. Thus, the order of dismissal should be reversed.

B. The claims improve analytics data processing technology

1. The district court erred by holding that "[PurePredictive] still cannot show that its claims improve the functioning of a computer-related technology rather than use computers as a tool." (Appx10:12-13) The district court did not consider the true claim limitations, how they are informed by the specification, or their benefits and advantages. Moreover, the "improvement" analysis under Alice is not limited to computer improvements, but extends to "improvement in any other technology or technical field." *Alice Corp. Pty. Ltd., v. CLS Bank Int'l*, 134 S. Ct. 2347, 2359 (2014).

2. As disclosed and described in the '446 specification, the inventions claimed by the '446 Patent are improvements to predictive analytics data processing technology and related computer technology at least in the following ways:

(i) The inventions claimed by '446 patent eliminate or reduce intervention by or needed input from data scientists, experts, and users. (Appx24, 1:42-44; Appx27, 7:14-16; Appx27, 7:29-31; Appx28, 10:8-10)

(ii) The inventions claimed by '446 patent improve the efficiency and effectiveness of the claimed predictive ensembles and determine a minimum

effective set of features for use in the predictive ensemble. (Appx29, 12:39-51; Appx28, 9:61-65; Appx30, 13:11-14; Appx32, 17:28-32)

(iii) The inventions claimed by '446 patent reduces noise from the resulting predictive ensembles. (Appx29, 12:55-59)

(iv) The inventions claimed by '446 patent optimize the overhead of the resulting predictive ensembles. (Appx29, 11:45-60)

(v) The inventions claimed by '446 patent improve and optimize confidence metrics. (Appx30, 13:19-27; Appx30, 13:35-39; Appx30, 14:9-25)

3. The predictive ensemble taught by the '446 Patent is also an improvement over other predictive-analytics modules, such as predictive voting systems. (Exhibit A to Mot. For Judicial Notice, 75-76:¶ 11) The predictive ensemble of the '446 Patent comparably has improved data processing time, reduced power consumption, and improved effectiveness of predictive data processing over voting system based predictive ensembles.

4. Thus, the order of dismissal should be reversed.

C. The appealed claims satisfy *Alice* step 2; the district court did not properly apply the "inventive concept" test

1. The district court erred in its application of the "inventive concept" test under *Alice* step 2. The district court's characterization of the claims as merely "recit[ing] the functional steps for collecting, analyzing, and refining data

through mathematical algorithms" is wrong and ignores the actual claim language. (Appx13:3-4)

2. The actual language of the claims-at-issue includes a predictive ensemble, which provides a structured data processing environment in which metadata is used to define relationships between classes or subsets of workload data and an array of different combined learned functions. The structured arrangement of the predictive ensemble is an "inventive concept" or "something more." *Alice*, 134 S. Ct. at 2355 (quoting *Mayo Collaborative Services v. Prometheus Labs., Inc.*, 132 S. Ct. 1289, 1294 (2012)). A predictive ensemble is not a mathematical algorithm.

3. But the dismissal order did not consider the actual language of each claim element either individually or in combination to see whether there was "something more". As with all of its other § 101 analysis, in each instance the district court erroneously only referenced its overbroad abstraction of the claims instead of the actual claim language. The district court's overbroad abstraction ignores the structured environment provided by the claimed predictive ensemble as described by the actual claim language. No claim language was ever addressed by the district court in this regard. Thus, core features such as the predictive ensemble were not meaningfully considered.

4. The actual claim language, considered in view of the specification, amply demonstrates that the claims are not directed to analyzing data through

mathematical algorithms. Claim 1 is directed to a predictive analytics factory *for forming a predictive ensemble*. Claim 14 is directed to a method for a predictive analytics factory *for forming a predictive ensemble.* Claim 17 is directed to a computer program product *for forming a predictive ensemble.* And claim 23 is directed to *a predictive ensemble.* Thus each of the claims-at-issue include patent eligible subject matter under § 101.

5. Accordingly, the order of dismissal should be reversed.

D. The district court erred by not applying the "clear and convincing" evidence standard under 35 U.S.C. § 282 and did not draw all reasonable inferences in favor of the plaintiff as required under Ninth Circuit precedent

1. The district court legally erred by not applying the "clear and convincing" evidence standard when it held the claims of the '446 Patent invalid. The district court also erred by not drawing all reasonable inferences from the alleged intrinsic evidence in favor of PurePredictive.

2. Challenges to patent validity under § 101 must satisfy a "clear and convincing" evidence standard. *See Microsoft Corp v. i4i Ltd. P'ship*, 564 U.S. 91, 103 (2011). "A patent shall be presumed valid." 35 U.S.C. § 282. The burden of establishing patent invalidity rests on the party asserting invalidity. *Id.*

3. The dismissal order is devoid of any reference to or discussion of 35 U.S.C. § 282, the presumption of patent validity, or the burden to establish invalidity by "clear and convincing" evidence. *See Microsoft Corp*, 564 U.S. at 103. (Appx2-

13) The district court's determination of invalidity did not take into account the presumption of patent validity by imposing the required "clear and convincing" evidence burden of proof on defendant.

4. Likewise, under Ninth Circuit precedent, the district court was required to "accept[] the plaintiff's allegations as true and draw[] all reasonable inferences in favor of the plaintiff." (Appx4:20-22). *See Usher v. City of Los Angeles*, 828 F.2d 556, 561 (9th Cir. 1987). But the district court did not meaningfully analyze the specification or intrinsic evidence by drawing all reasonable inferences in favor of PurePredictive. Rather, the district court accepted defendant's conclusory statement regarding the meaning of the claims.

5. Thus, the order of dismissal should be reversed.

E. The district court erroneously held all claims invalid; amendment of the Complaint would not be futile

1. The district court erroneously held that "[a]mendment [of the Complaint] would be futile in light of the analysis above." (Appx13:12) Thus the dismissal order essentially holds all of the '446 claims, including dependent claims, invalid without considering any of the claim language covering core features of the claimed invention.

2. PurePredictive's infringement cause of action was not intended to be limited to infringement of the independent claims. Indeed, it broadly alleges that defendant "infringes one or more claims" of the '446 Patent. (Appx48:10-11)

3. Dismissing the Complaint without considering the dependent claims is contrary to the plain language of 35 U.S.C. § 282, which states: "Each claim of a patent . . . shall be presumed valid independently of the validity of other claims." 35 U.S.C. § 282.

4. Defendant also infringes a number of dependent claims, including for example, claim 12. Claim 12's language is directed to the orchestration module for directing workload data through the structured environment of the predictive ensemble, and in particular to the classification of data and providing a confidence metric for said classification. Thus claim 12 is directed to the transformation of data by the predictive ensemble, which is clearly patent eligible under § 101. *In re Bilski*, 545 F.3d at 963.

5. Thus, the order of dismissal should be reversed.

ARGUMENT

I. LEGAL STANDARD OF REVIEW

A. Dismissal under Rule 12(b)(6)

This Court applies the law of the regional circuit when reviewing a district court's grant of a motion to dismiss under Rule 12(b)(6) for failure to state a claim upon which relief can be granted. *K-Tech Telecomms., Inc. v. Time Warner Cable, Inc.*, 714 F.3d 1277, 1282 (Fed. Cir. 2013). The Ninth Circuit reviews *de novo* a

district court's dismissal for failure to state a claim pursuant to Rule 12(b)(6). *See Knievel v. ESPN*, 393 F.3d 1068, 1072 (9th Cir. 2005).

In deciding motions to dismiss, courts look at the allegations of the complaint to determine if "enough facts to state a claim to relief that is plausible on its face" are alleged, *Bell Atl. Corp. v. Twombly*, 550 U.S. 544, 570 (2007). A claim is facially plausible "when the plaintiff pleads factual content that allows the court to draw the reasonable inference that the defendant is liable for the misconduct alleged." *Ashcroft v. Iqbal*, 556 U.S. 662, 678 (2009). In deciding whether the plaintiff has stated a claim upon which relief can be granted, the court accepts the plaintiff's allegations as true and draws all reasonable inferences in favor of the plaintiff. *See Usher*, 828 F.2d at 561.

B. Patent Validity

"A patent shall be presumed valid. Each claim of a patent . . . shall be presumed valid independently of the validity of other claims; dependent or multiple dependent claims shall be presumed valid even though dependent upon an invalid claim." 35 U.S.C. § 282. The burden of establishing patent invalidity rests on the party asserting invalidity. *Id.*

Challenges to patent validity under § 101 must satisfy a "clear and convincing" evidence standard. *See Microsoft*, 564 U.S. at 103 (holding that the presumption of validity under 35 U.S.C. § 282 requires the clear and convincing

evidence standard to apply to all validity challenges because "we must presume that Congress intended to incorporate the heightened standard of proof, unless the statute otherwise dictates" and nothing "suggests that Congress meant to . . . enact a standard of proof that would rise and fall with the facts of each case" (internal quotes omitted)); *CLS Bank*, 717 F.3d at 1304-05 *aff'd* 134 S. Ct. 2347 (2014); *DataTern, Inc. v. Microstrategy, Inc.*, 2015 WL 5190715 at *8–9 (applying the presumption of validity and the clear-and-convincing standard in § 101 analysis).

C. Claim Construction

The ultimate question of proper claim construction is a question of law that this Court reviews *de novo*. *Teva Pharm. USA, Inc. v. Sandoz, Inc.* 135 S. Ct. 831, 837 (2015). Likewise, "when the district court reviews only evidence intrinsic to the patent (the patent claims and specifications, along with the patent's prosecution history), the judge's determination will amount solely to a determination of law" which will be reviewed *de novo*. *Id.* at 841. But subsidiary factual findings based on extrinsic evidence are reviewed under the clearly erroneous standard. *Id.* at 838.

D. Patent eligibility under § 101

This Court reviews determinations of patent eligibility under 35 U.S.C. § 101 *de novo*. *McRO, Inc. v. Bandai Namco Games America Inc.*, 837 F.3d 1299, 1311 (Fed. Cir. 2016); *DDR Holdings, LLC v. Hotels.Com, L.P.*, 773 F.3d 1245, 1255 (Fed. Cir. 2014).

Section 101 dictates that a patent may be obtained for "any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof." 35 U.S.C. § 101. There are only three narrow, judicially-created, exceptions to the broad language of 35 U.S.C. § 101 – "laws of nature, natural phenomena, and abstract ideas." *Diamond v. Diehr*, 450 U.S. 175, 185 (1981); *see also Bilski v. Kappos*, 561 U.S. 593, 601-02 (2010) ("*Bilski II*"). But the Supreme Court has emphasized caution when applying these three narrow exceptions: "[W]e tread carefully in construing this exclusionary principle, lest it swallow all of patent law. At some level, 'all inventions . . . embody, use, reflect, rest upon, or apply laws of nature, natural phenomena, or abstract ideas.'" *Alice*, 134 S. Ct. at 2354 (quoting *Mayo*, 132 S. Ct. at 1293 (internal citation omitted)).

This Court has held that a patent claim should not be found ineligible as directed to an abstract idea under Section 101 unless the abstractness "exhibit[s] itself so manifestly as to override the broad statutory categories of eligible subject matter." *Research Corp. Techs., Inc. v. Microsoft Corp.*, 627 F.3d 859, 868 (Fed. Cir. 2010) (emphasis added).

II. THE CLAIMS COVER PATENT ELIGIBLE-SUBJECT MATTER

The district court erred by granting the motion to dismiss holding that claims 1, 14, 17, and 23 are patent ineligible under § 101.

The claims-at-issue are directed to an improvement in the technology of predictive analytics data processing. Each of the claims includes a predictive ensemble that provides a structured environment for processing analytics data. The predictive ensemble comprises a combined arrangement of learned functions, a synthesized metadata rules set, and means for implementing the metadata rules set. The workload environment for predictive analytics data processing is structured in the predictive ensemble using the synthesized metadata rules set, which defines the division of data processing workload. The synthesized metadata rules set dictates which subsets of data get processed by which of the different learned functions within the predictive ensemble. As set forth in the last element of claim 1:

> the predictive ensemble comprising a subset of multiple learned functions from the plurality of learned functions, the multiple learned functions selected and combined based on the evaluation metadata for the plurality of learned functions, the predictive ensemble comprising a rule set synthesized from the evaluation metadata to direct data through the multiple learned functions such that different learned functions of the ensemble process different subsets of the data based on the evaluation metadata.

(Appx35, 23:8-18) (emphasis added)

But the district court described the '446 Patent's claims at such a high level of abstraction and untethered from the claim language that it ensured the claims would be patent ineligible. The district court's order of dismissal also erred by finding the '446 Patent claims were invalid without applying a "clear and convincing" evidence standard or giving deference to the patent's presumption of validity. Likewise, the

district court erred in not according the best light to the ‘446 Patent. The claims-at-issue are directed to an improvement in the technology of predictive analytics data processing.

The Supreme Court clarified the test for the subject matter eligibility of claims under 35 U.S.C. § 101 in *Mayo*. In *Alice*, the Court summarized the *Mayo* standard: “[f]irst, we determine whether the claims at issue are directed to one of those patent-ineligible concepts. If so, we then ask, ‘what else is there in the claims before us?’ To answer that question, we consider the elements of each claim both individually and ‘as an ordered combination’ to determine whether the additional elements ‘transform the nature of the claim’ into a patent-eligible application.” *Alice*, 134 S. Ct. at 2355 (*quoting Mayo*, 132 S. Ct. at 1297-98). The Supreme Court has “described step two of this analysis as a search for an ‘inventive concept’—i.e., an element or combination of elements that is ‘sufficient to ensure that the patent in practice amounts to significantly more than a patent upon the [ineligible concept] itself.’” *Alice*, 134 S. Ct. at 2355 (quoting *Mayo*, 132 S. Ct. at 1294) (alteration in original).

Without limiting other eligible subject matter under § 101, the Supreme Court has indicated that patent claims satisfy § 101’s requirements if they “improve the functioning of the computer itself,” or “effect an <u>improvement in any other technology or technical field</u>.” *Alice*, 134 S. Ct. at 2359 (emphasis added). And this

Court held in *DDR Holdings* that that patent claims satisfied § 101 when "the claimed solution is necessarily rooted in computer technology in order to overcome a problem specifically arising in the realm of computer networks." *DDR Holdings*, 773 F.3d at 1257.

In *Enfish*, this Court stated that "the first step of the [Alice] inquiry is a meaningful one, *i.e.*, that a substantial class of claims are *not* directed to a patent-ineligible concept." *Enfish*, 822 F.3d at 1335 (emphasis in original). The "directed to" inquiry, therefore,

> cannot simply ask whether the claims involve a patent-ineligible concept, because essentially every routinely patent-eligible claim involving physical products and actions involves a law of nature and/or natural phenomenon - after all, they take place in the physical world. See Mayo, 132 S. Ct. at 1293 ("For all inventions at some level embody, use, reflect, rest upon, or apply laws of nature, natural phenomena, or abstract ideas.") Rather, the "directed to" inquiry applies a stage-one filter to claims, <u>considered in light of the specification</u>, based on whether "their character as a whole is directed to excluded subject matter." (*Internet Patents Corp. v. Active Network, Inc.*, 790 F.3d 1343, 1346 (Fed. Cir. 2015).

Id. (emphasis added)

Further, this Court has stated: "[w]e therefore look to whether the claims in these patents focus on a specific means or method that <u>improves the relevant technology</u> or are instead directed to a result or effect that itself is the abstract idea and merely invoke generic processes and machinery." *McRO*, 837 F.3d at 1314

(citing *Enfish*, 822 F.3d at 1336 and *Rapid Litig. Mgmt. Ltd. v. CellzDirect, Inc.*, 827 F.3d 1042, 1048 (Fed. Cir. 2016)) (emphasis added).

The asserted claims focus on an improvement to the technology of predictive analytic data processing. Each of the claims include a predictive ensemble having a structured data processing environment and arrangement of combined learned functions where different subsets of data are directed to different learned functions for predictive analytics data processing. The meaning of the claims was not properly considered in light of the '446 specification. Thus, the district court's order of dismissal should be reversed.

A. The district court oversimplified the claims without accounting for their core features

The district court oversimplified the claims in its § 101 analysis by finding they were directed to "the abstract concept of the manipulation of mathematical functions and make use of computers only as tools, rather than provide a specific improvement on a computer-related technology" (Appx2:18-20), "the patent-ineligible abstract concept of testing and refining mathematical algorithms" (Appx11:3-4), or merely "recite the functional steps for collecting, analyzing, and refining data through mathematical algorithms." (Appx13:3-4) Describing the claims at such a high level of abstraction and untethered from the actual language of the claims, as informed by the specification, all but ensures that the "abstract idea" exceptions to § 101 will swallow the rule:

> Any claim can be stripped down, simplified, generalized, or paraphrased to remove all of its concrete limitations, until at its core, something that could be characterized as an abstract idea is revealed. Such an approach would 'if carried to its extreme, make all inventions unpatentable because all inventions can be reduced to underlying principles of nature which, once known, make their implementation obvious.

CLS Bank, 717 F.3d at 1298 *aff'd* 134 S. Ct. *2347 (2014) (quoting Diamond*, 450 U.S. at 189 n.12).

The dismissal order stripped down and generalized the claims. It includes no meaningful discussion regarding the actual claim language or the meaning of the claim terms considered in view of the specification. Rather, the district court merely quoted claim 14 and then makes the following conclusory statement:

> Put very simply, the method performs predictive analytics in three steps. First, it receives data and generates "learned functions," or, for example, regressions from that data. See '446 Patent at 8:66–9:12. Second, it evaluates the effectiveness of those learned functions at making accurate predictions based on the test data. Finally, it selects the most effective learned functions and creates a rule set for additional data input. These three steps comprise the predictive analytics factory's method. Claim 1 recites a module-based apparatus for this predictive analytics factory, Compl. ¶ 16, Claim 17 recites a computer program product to perform the operations of the predictive analytics factory, Compl. ¶ 18, and Claim 23 recites a predictive analytics ensemble, Compl. ¶ 19.

(Appx3:15-24)

This overbroad abstraction does not account for core features of the claims. The order does not include any discussion of the actual claim language. The claims at issue are more specific than the dismissal order's oversimplified rephrase. As a

whole they are not directed to excluded subject matter such as a common business method or a fundamental practice long prevalent that uses a general computer. Notably, each of the claims includes a predictive ensemble consistent with the inventions disclosed in the '446 Patent. The predictive ensemble of the '446 Patent provides a structured environment for predictive analytics data processing. Yet the district court's erroneous abstraction of the claims informed the rest of the district court's findings without reference to the claim language. The district court never discussed the claim language regarding the predictive ensemble feature.

For example, when evaluating whether or not the claims were directed to patent ineligible subject matter, the district court addressed the claims solely by reference to the court abstracted "steps" recited above. No further discussion of the claim language was conducted.

The dismissal order addressed the court-created "first step" (on page 8 of Order) as follows: "The first step, generating learned functions or regressions from data—the basic mathematical process of, for example, regression modeling, or running data through an algorithm—is not a patentable concept." (Appx9:8). No reference was made to any claim language. No mention was made of the claimed predictive ensemble, which provides a structured environment for data processing. Likewise, the dismissal order addressed the court-created second and third "steps" (on page 9 of Order) as follows:

> The next steps of the method are similarly abstract. The method takes the learned functions, evaluates their effectiveness, and selects those most effective to create a rule set. These are mathematical processes that not only could be performed by humans but also go to the general abstract concept of predictive analytics rather than any specific application.

(Appx10:1-4) Again, no reference was made to any claim language. And the Order did not include any discussion of the claimed predictive ensemble.

Notable was the district court's focus on whether the claims were directed to patent ineligible subject matter rather than whether the "claims are *not* directed to a patent-ineligible concept" as described in *Enfish*, 822 F.3d at 1335 (emphasis in original). (Appx6-11) In other words, the district court's focus was to search for patent ineligible subject matter rather than search for patent eligible subject matter. This approach ensured the court would find ineligible subject matter while overlooking core features of the claims. *CLS Bank*, 717 F.3d at 1298 *aff'd* 134 S. Ct. 2347 (2014) (quoting *Diamond*, 450 U.S. at 189 n.12 ("Any claim can be stripped down, simplified, generalized, or paraphrased to remove all of its concrete limitations, until at its core, something that could be characterized as an abstract idea is revealed.")).

As noted above, the district court overlooked core features found in the claims. Each of the asserted claims includes language covering a predictive ensemble. A predictive ensemble is not a mathematical formula. Instead, it is a computer implemented module that provides a structured environment for processing

predictive analytics data. As illustrated in FIG. 4 of the '446 Patent (below), the predictive ensemble includes (i) an orchestration module, (ii) a synthesized metadata rule set, and (iii) a combination or arrangement of synthesized learned functions.

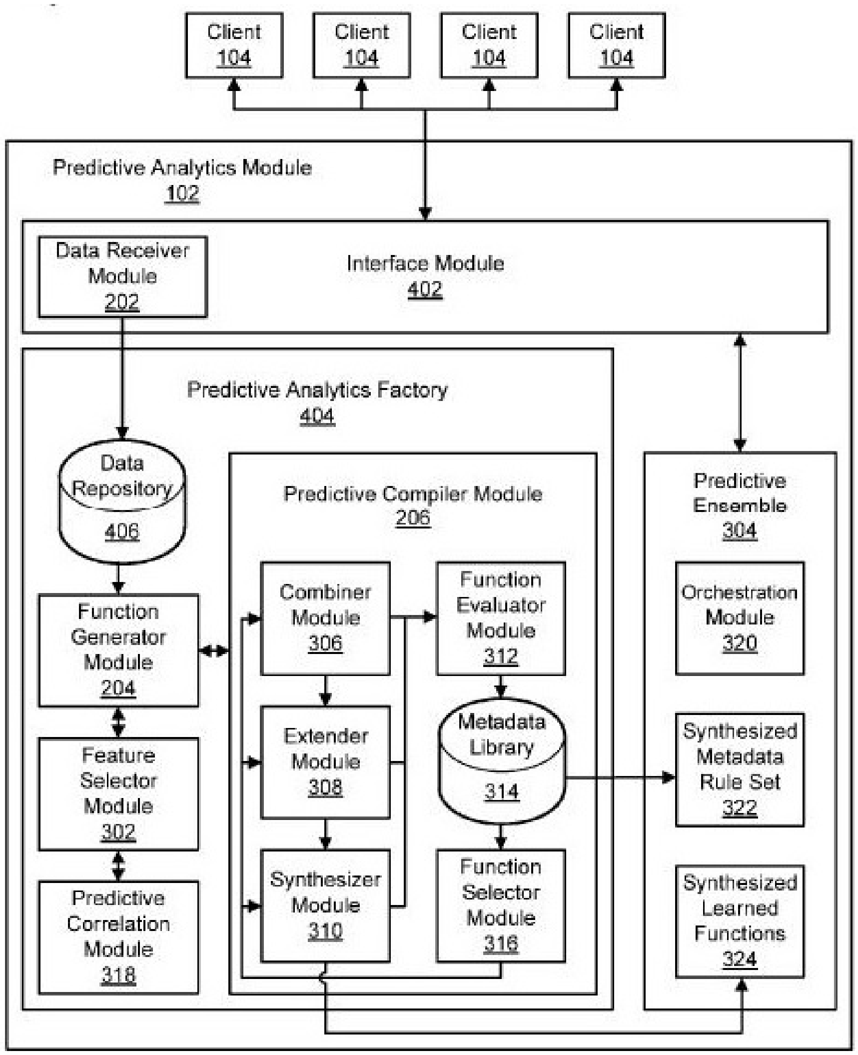


FIG. 4

In the predictive ensemble, different learned functions are synthesized and combined into a structured predictive data processing environment. As described in the patent (col. 20, ll. 54-62), the set of learned functions may include:

> different classes [of learned functions] including a collection of decision trees 502a, configured to receive or process a subset A-F of the feature set of the predictive ensemble 304, a collection of support vector machines ("SVMs") 502b with certain kernels and with an input space configured with particular subsets of the feature set G-L, and a selected group of regression models 502c, here depicted as a suite of single layer ("SL") neural nets trained on certain feature sets K-N.

(Appx33, 20:54-62)

The analytics data processing workload is organized or structured among the different synthesized learned functions using the synthesized metadata rule set. As described in the ‘446 Patent (col. 18, ll. 23-28):

> the synthesized metadata rule set 322 comprises a set of rules or conditions from the evaluation metadata of the metadata library 314 that indicate to the orchestration module 320 which features, instances, or the like should be directed to which synthesized learned function 324.
>
> (Appx32, 18:23-28)

The synthesized metadata rule set is generated using a synthesizer module. As described in the ‘446 Patent (col. 18, ll. 33-39):

> The synthesizer module 310 may use that evaluation metadata to determine rules for the synthesized metadata rule set 322, indicating which features, which instances, or the like the orchestration module 320 the orchestration module 320 [sic] should direct through which learned functions, in which order, or the like.
>
> (Appx32, 18:33-39)

Thus, the metadata rule set defines the direction or flow for the workload data and dictates the division of work load among learned functions. Different learned

functions process different subsets of data. The orchestration module then directs the appropriate class or subset of data to the appropriate learned function and in the appropriate order based on the metadata rule set. As described in the '446 Patent (col. 18, ll. 11-22):

> The orchestration module 320 . . . is configured to direct workload data through the predictive ensemble 304 to produce a result, such as a classification, a confidence metric, . . . an answer, a prediction, a recognized pattern, a rule, [and/or] a recommendation . . . . [T]he orchestration module 320 uses evaluation metadata from the function evaluator module 312 and/or the metadata library 314, such as the synthesized metadata rule set 322, to determine how to direct workload data through the synthesized learned functions 324 of the predictive ensemble 304.

(Appx32, 18:11-22)

Thus, the predictive ensemble of the '446 Patent provides a structured data processing environment in which different subsets of workload data are assigned to take different data processing paths to be processed by different combined learned functions.

Each of the asserted claims covers a predictive ensemble consistent with the specification. For example, claims 1 and 14 include, in part, the following language,

> <u>the predictive ensemble comprising a subset of multiple learned functions</u> from the plurality of learned functions, the multiple learned functions selected and <u>combined based on the evaluation metadata</u> for the plurality of learned functions, the predictive ensemble comprising a rule set synthesized from <u>the evaluation metadata to direct data through the multiple learned functions such that different learned functions of the ensemble process different subsets of the data based on the evaluation metadata</u>.

(Appx35, 23:8-18) (emphasis added)

But the overbroad abstraction of the claims by the district court overlooks the specific claim language and in particular this core feature. The predictive ensemble feature is comparable to the "self-referential" database model held as patent eligible in *Enfish*. *Enfish*, 822 F.3d at 1330-33, 1346. In *Enfish*, the patents-at-issue covered a logical model for a computer database that defined how the various elements of information in the database are related to each other. Enfish's logical model includes all data entities in a single table, with column definitions provided by rows in that same table. *Id.* at 1330. The patents describe this as the "self-referential" property of the database. This Court held that claims covering this "self-referential" feature were patent eligible under § 101.

The predictive ensemble feature claimed in the '446 Patent likewise is a structured environment that defines how various elements of information (e.g., subsets of data) are related to various combined learned functions. It is not a mathematical formula. Rather, metadata is used to define relationships between different subsets of workload data and different synthesized learned functions. The combined arrangement (e.g., assembly or structure) of synthesized learned functions has a structured relationship with respect to subsets of data to be processed. The synthesized metadata rules set defines the relationship between the different subsets of data and the different combined learned functions. (Appx35, 23:8-18) Accordingly, the asserted claims are directed to patent eligible subject matter under § 101.

Furthermore, asserted claims covering a process for forming a predictive ensemble of the '446 Patent are patent eligible because they cover a transformation. Before the Supreme Court established the *Alice/Mayo* framework, this Court used "the machine-or-transformation test" when "determining patent eligibility of a process under § 101." *In re Bilski*, 545 F.3d at 956, *aff'd sub nom. Bilski v. Kappos*, 561 U.S. 593 (2010). Under that test, a process is patent-eligible if: "(1) it is tied to a particular machine or apparatus, *or* (2) it transforms a particular article into a different state or thing." *In re Bilski*, 545 F.3d at 954 (emphasis added). Although it "is not the sole test" for patent-eligibility, the machine-or-transformation test remains "a useful and important clue." *Bilski*, 561 U.S. at 604. The claims here satisfy that test.

For example, claim 14 is directed to the method for a predictive analytics factory *for forming a predictive ensemble* consistent with the '446 specification. Claim 14 states:

> 14. A <u>method for a predictive analysis factory</u>, the method comprising:
>
> pseudo-randomly generating a plurality of learned functions based on training data without prior knowledge regarding suitability of the generated learned functions for the training data, the training data received <u>for forming a predictive ensemble</u> customized for the training data;
>
> evaluating the plurality of learned functions using test data to generate evaluation metadata indicating an effectiveness of different learned functions at making predictions based on different subsets of the test data; and

> <u>forming the predictive ensemble comprising</u> a subset of multiple learned functions from the plurality of learned functions, the subset of multiple learned functions selected and combined based on the evaluation metadata, the predictive ensemble comprising a rule set synthesized from the evaluation metadata to direct different subsets of the workload data through different learned functions of the multiple learned functions based on the evaluation metadata.

(Appx35, 24:21-41) (emphasis added)

Because the invention transforms pseudo-randomly generated learned functions and metadata into the structured data processing environment of the '446 predictive ensemble, as explained above, the claimed process is patent eligible. *In re Bilski*, 545 F.3d at 963; *see In re Abele*, 684 F.2d 902 (C.C.P.A. 1982) (upholding invention that takes X-ray attenuation data and displays it in visual form on a screen). Likewise, claims 1 and 17 are respectively directed to a predictive analytics factory and a computer program product that transform pseudo-randomly generated learned functions and metadata into the structured data processing environment of the '446 predictive ensemble. Accordingly, they are likewise patent eligible.

Thus, the order of dismissal should be reversed.

B. The claims improve analytics data processing technology

The district court erred by holding that "[PurePredictive] still cannot show that its claims improve the functioning of a computer-related technology rather than use computers as a tool." (Appx10:12-13) The district court did not consider the true claim limitations, how they are informed by the specification, or their benefits and

advantages. Moreover, the "improvement" analysis under *Alice* is not limited to computer improvements, but extends to "improvement in any other technology or technical field." *Alice*, 134 S. Ct. at 2359.

The '446 specification teaches the invention improves predictive analytics data processing technology. Among improvements disclosed in the specification, are:

1. Generation of effective predictive ensembles without intervention of data scientists, experts, or users. The '446 specification states the following about the invention's elimination or reduction of input from data scientists, experts, and users:

> a predictive analytics factory configured to generate a predictive ensemble regardless of the particular field or application, with little or no input from a user or expert.

(Appx24, 1:42-44)

> The predictive analytics module 102 . . . generates predictive ensembles for the clients 104, with little or no input from a Data Scientist or other expert

(Appx27, 7:14-16)

> provide[s] predictive ensembles that are customized and finely tuned for data from a specific client 104, without excessive intervention or fine-tuning

(Appx27, 7:29-31)

> eliminates or minimizes the role of a Data Scientist or other expert in generation of a predictive ensemble

(Appx28, 10:8-10)

2. <u>More effective and efficient predictive ensembles</u>. The '446 specification states the following about how the invention improves the efficiency and effectiveness of predictive ensembles:

> the predictive compiler module 206 . . . may iteratively increase the number of features used to generate predictive ensembles 304 until an increase in effectiveness or usefulness of the results of the generated predictive ensembles 304 fails to satisfy a feature effectiveness threshold. By increasing the number of features until the increases stop being effective . . . the predictive compiler module 206 may determine a minimum effective set of features for use in a predictive ensemble 304, so that generation and use of the predictive ensemble 304 is both effective and efficient.

(Appx29, 12:39-51)

> the function generator module 204 ensures that at least a subset of the generated learned functions, either individually or in combination, are useful, suitable, and/or effective for the training data without careful curation and fine tuning by a Data Scientist or other expert.

(Appx28, 9:61-65)

> At each iteration, the function evaluator module 312 may determine an overall effectiveness of the learned functions in aggregate for the current iteration's selected combination of features.

(Appx30, 13:11-14)

> the metadata library 314 may index evaluation metadata by learned function, by feature, by instance, by training data, by test data, by effectiveness, and/or by another category 30 or attribute and may provide query access to the indexed evaluation metadata.

(Appx32, 17:28-32)

3. Excludes or reduces noise from predictive ensembles. The '446 specification states the following about how the invention reduces noise from predictive ensembles:

> [O]nce the feature selector module 302 determines that a feature is merely introducing noise, the predictive compiler module 206 excludes the feature from future iterations, and from the predictive ensemble 304.

(Appx29, 12:55-59)

4. Optimizes overhead of predictive ensembles. The '446 specification states the following about how the invention optimizes the overhead of predictive ensembles:

> the feature selector module 302 determines which features of initialization data to use in the predictive ensemble 304, and in the associated learned functions, and/or which features of the initialization data to exclude from the predictive ensemble 304, and from the associated learned functions. As described above, initialization data, and the training data and test data derived from the initialization data, may include one or more features. Learned functions and the predictive ensembles 304 that they form are configured to receive and process instances of one or more features. Certain features may be more predictive than others, and the more features that the predictive compiler module 206 processes and includes in the generated predictive ensemble 304, the more processing overhead used by the predictive compiler module 206, and the more complex the generated predictive ensemble 304 becomes. 304.

(Appx29, 11:45-60)

5. Higher and optimized confidence metrics. The '446 specification states the following about how the invention improves or optimizes confidence metrics:

> the predictive correlation module 318 determines one or more features, instances of features, or the like that correlate with higher confidence metrics (e.g., that are most effective in predicting results with high confidence). The predictive correlation module 318 may cooperate with, be integrated with, or otherwise work in concert with the feature selector module 302 to determine one or more features, instances of features, or the like that correlate with higher confidence metrics.

(Appx30, 13:19-27)

> The predictive correlation module 318 . . . is configured to harvest metadata regarding which features correlate to higher confidence metrics, to determine which feature was predictive of which outcome or result, or the like.

(Appx30, 13:35-39)

> In determining features that are predictive, or that have a greatest contribution to a predicted result or confidence metric, the predictive correlation module 318 may balance a frequency of the contribution of a feature and/or an impact of the contribution of the feature. For example, a certain feature or set of features may contribute to the predicted result or confidence metric frequently, for each instance or the like, but have a low impact. Another feature or set of features may contribute relatively infrequently, but has a very high impact on the predicted result or confidence metric (e.g. provides at or near 100% confidence or the like). While the predictive correlation module 318 is described herein as determining features that are predictive or that have a greatest contribution, in other embodiments, the predictive correlation module 318 may determine one or more specific instances of a feature that are predictive, have a greatest contribution to a predicted result or confidence metric, or the like.

(Appx30, 14:9-25)

The predictive ensemble taught by the ‘446 Patent is also an improvement over other predictive analytics modules, such as predictive voting systems. The distinction was made during prosecution of the ‘446 Patent. In an April 8, 2014

Office Action, the Patent Office rejected a number of the claims under 35 U.S.C. 103(a) referencing US 2008/0162487 A1 by Richter ("*Richter*"). (Exhibit A to Mot. For Judicial Notice, 42:¶ 6) *Richter* disclosed a voting system based ensemble. In a June 11, 2014 After Final Amendment and Response to Office Action (pgs. 15-16), the '446 Patent applicant stated:

> The present amendments clarify that the synthesized rule set directs data such that different learned functions of the ensemble process different subsets of the data, such as different features/columns of data, different rows/instances of data, or both. For example, Fig. 5 of the Application depicts one embodiment of a predictive ensemble where different learned functions (e.g., learned functions 502, combined learned functions 504, and extended learned functions 506) process different subsets of data (e.g., different features/columns of data rows/instances, Features A-F, Features G-L, Features K-N, Features N-S, Features T-U, Features L-R). The ensemble methods mentioned in Fig. 13 of Richter, however, "voting," "average," "weighted average," and "hueristic rules" [sic.] do not distinguish between learned functions, with all learned functions processing all data. *Richter* does not teach any synthesized rules for directing data through multiple learned functions of an ensemble so that different learned functions process different subsets of the data.

(Exhibit A to Mot. For Judicial Notice, 75-76:¶ 11)

The distinction between the '446 predictive ensemble and a voting system reflects a significant improvement in the predictive analytics data processing technology. The division of workload data-processing labor by the predictive ensemble of the '446 Patent, contrasted with other predictive analytics modules that employ concurrent processing of the same data by each learned function, has a number of advantages. Predictive analytics data will get processed quicker. Less

power will be required to process the data. And classes or subsets of workload data will get processed by the learned functions most suited for the data. Thus, the predictive ensemble of the '446 Patent improves data processing time, reduces power consumption, and improves effectiveness of predictive data processing.

This is not a case of a "fundamental economic and conventional business practice" or a "well-known method of organizing human behavior" being applied to an ordinary computer as a tool. The claims are not abstract because they improve analytics data processing technology and computer functionality.

Accordingly, the dismissal order finding the asserted claims invalid under § 101 should be reversed.

C. The appealed claims satisfy *Alice* step 2; the district court did not properly apply the "inventive concept" test

The district court erred in its application of the "inventive concept" test under *Alice* step 2. Even if the appealed claims were directed to an abstract idea under *Alice* step 1 – which they are not – the claims satisfy the "something more" or "inventive concept" of *Alice* step 2.

The *Alice* step 2 analysis entails a "search for an inventive concept—*i.e.*, an element or combination of elements that is sufficient to ensure that the patent in practice amounts to significantly more than a patent upon the [ineligible concept] itself." *Alice*, 134 S. Ct. at 2355 (quoting *Mayo*, 132 S. Ct. at 1294) (internal quotation marks and citations omitted). As noted in *Bascom*:

> The inventive concept inquiry requires more than recognizing that each claim element, by itself, was known in the art. As is the case here, an inventive concept can be found in the non-conventional and non-generic arrangement of known, conventional pieces.

Bascom Global Internet Services, Inc. v. AT&T Mobility LLC, 827 F.3d 1341, 1350 (Fed. Cir. 2016).

In its *Alice* step 2 analysis, the district court did not address the claim language or properly consider the meaning of any specific element of any asserted claim, particularly in view of the '446 specification. As discussed above, the district court overly simplified the claims to abstract core features out of the claims. The district court then erroneously used this overbroad abstraction of the claims for each analysis, including its *Alice* step 2 analysis, instead of referring to the actual claim language. For example, when evaluating whether an inventive concept could be found in a combination of the elements of the asserted claims, the district court's analysis and finding consisted of:

> While [PurePredictive] claims that "the ordered combination of the claims" provides an inventive concept, there is nothing inventive about its particular arrangement. Instead, its claims recite the functional steps for collecting, analyzing, and refining data through mathematical algorithms. . . . [PurePredictive]'s technology, while perhaps an effective method, is simply an implementation of the basic concept of predictive analytics on an apparatus, computer program product, or other medium.

(Appx13:1-7)

Again, the district court's overbroad abstraction wholly ignores core features of the asserted claims, including the structured environment provided by the claimed predictive ensemble (which provides division of labor among combined synthesized learned functions and directs different subsets of data to different synthesized learned functions for analytics data processing). No claim language was ever addressed by the district court in this regard.

The district court's characterization of the claims as merely "recit[ing] the functional steps for collecting, analyzing, and refining data through mathematical algorithms" is wholly inaccurate and ignores the language of the claims. (Appx13:3-4) Claim 1, for example is directed to a predictive analytics factory *for forming a predictive ensemble* consistent with the '446 Patent. It is not directed to analyzing data through mathematical algorithms.

Claim 1 states, in relevant part:

> 1. An apparatus for <u>a predictive analytics factory</u>, the apparatus comprising:
>
> a receiver module configured to receive training data <u>for forming a predictive ensemble</u> customized for the training data;
>
> a function generator module configured to pseudo-randomly generate a plurality of learned functions based on the training data without prior know ledge regarding suitability of the generated learned functions for the training data;
>
> a function evaluator module configured to perform an evaluation of the plurality of learned functions using test data and to maintain evaluation metadata for the plurality of learned functions, the

> evaluation metadata comprising one or more of an indicator of a training data set used to generate a learned function and an indicator of one or more decisions made by a learned function during the evaluation; and
>
> a predictive compiler module <u>configured to form the predictive ensemble, the predictive ensemble comprising</u> a subset of multiple learned functions from the plurality of learned functions, the multiple learned functions selected and combined based on the evaluation metadata for the plurality of learned functions, the predictive ensemble comprising a rule set synthesized from the evaluation metadata to direct data through the multiple learned functions such that different learned functions of the ensemble process different subsets of the data based on the evaluation metadata.

(Appx34, 22:56 – Appx35, 23:18) (emphasis added)

It is clear from the language of claim 1 that is directed to a predictive analytics factory *for forming a predictive ensemble*. The predictive ensemble of the ‘446 Patent includes combined learned functions, but is not a mathematical algorithm. Rather a predictive ensemble of the ‘446 Patent is a predictive analytics module providing a structured environment for predictive data processing. It is an arrangement of combined learned functions in which relationships between subsets of data and combined learned functions are defined by a metadata rules set. Thus, different data subsets may be directed to different learned functions within the predictive ensemble based on the metadata. Accordingly, the district court’s characterization of claim 1 is erroneous.

Similarly, and as discussed above, claim 14 is directed to the method for a predictive analytics factory *for forming a predictive ensemble* consistent with the

'446 specification. It is not directed to "the collection, analyzing, and refining of data through a mathematical formulation." As noted above, claim 14 states:

> 14. A method for <u>a predictive analysis factory</u>, the method comprising:
>
> pseudo-randomly generating a plurality of learned functions based on training data without prior knowledge regarding suitability of the generated learned functions for the training data, the training data received <u>for forming a predictive ensemble</u> customized for the training data;
>
> evaluating the plurality of learned functions using test data to generate evaluation metadata indicating an effectiveness of different learned functions at making predictions based on different subsets of the test data; and
>
> <u>forming the predictive ensemble comprising</u> a subset of <u>multiple learned functions</u> from the plurality of learned functions, the subset of multiple learned functions selected and <u>combined based on the evaluation metadata</u>, <u>the predictive ensemble comprising</u> a rule set synthesized from the <u>evaluation metadata to direct different subsets of the workload data through different learned functions of the multiple learned functions based on the evaluation metadata</u>.

(Appx35, 24:21-41) (emphasis added)

Similarly, claim 17 is directed to a computer program product *for forming a predictive ensemble* consistent with the '446 specification. It is not directed to "the collection, analyzing, and refining of data through a mathematical formulation." Claim 17 states:

> 17. <u>A computer program product</u> comprising a non-transitory computer readable storage medium storing computer usable program code executable to perform operations for a predictive analysis factory, the operations comprising:

> pseudo-randomly determining a plurality of learned functions using training data without prior knowledge regarding suitability of the determined learned functions for the training data, the training data comprising a plurality of features, the training data received for forming a predictive ensemble customized for the training data;
>
> selecting a subset of the features of the training data based on evaluation metadata generated for the plurality of learned functions, the evaluation metadata comprising an effectiveness metric for a learned function; and
>
> forming the predictive ensemble, the predictive ensemble comprising at least two learned functions from the plurality of learned functions, the at least two learned functions using the selected subset of features, the at least two learned functions selected and combined based on the evaluation metadata, the predictive ensemble comprising a rule set synthesized from the evaluation metadata to direct data through the at least two learned functions so that different learned functions process different features of the selected subset of features.

(Appx35, 24:54 – Appx36, 25:10) (emphasis added)

The language of claim 23 is also strikingly inconsistent with the holding of the district court. Claim 23 is directed to *a predictive ensemble* consistent with the '446 specification. Again, a predictive ensemble is not a mathematical algorithm. Rather it provides a structured environment for predictive analytics data processing in which the relationship between subsets of data and combined learned functions is defined by a metadata rules set. Thus, claim 23 is not directed to "the collection, analyzing, and refining of data through a mathematical formulation." Claim 23 states:

> 23. A predictive analytics ensemble comprising:
>
> multiple learned functions synthesized from a larger plurality of learned functions, the multiple learned functions selected and combined based on evaluation metadata for an evaluation of the larger plurality of learned functions, wherein the larger plurality of learned functions are generated pseudo-randomly from training data without prior knowledge of a suitability of the larger plurality of learned functions for the training data;
>
> a metadata rule set synthesized from the evaluation metadata for the plurality of learned functions for directing data through different learned functions of the multiple learned functions to produce a result; and
>
> an orchestration module configured to direct the data through the different learned functions of the multiple learned functions based on the synthesized metadata rule set to produce the result.

(Appx36, 26:4-19) (emphasis added).

Thus, the district court also erred by ignoring the language of the claims and applying an overly broad abstraction of the claims without reference to core features of the claimed invention. The district court also erred by not considering the actual claim language when considering whether the claims included an "inventive concept" under the *Alice* step 2 analysis.

Accordingly, the dismissal order holding the asserted claims invalid should be reversed.

D. The district court did not apply the "clear and convincing" evidence standard under 35 U.S.C. § 282 or draw all reasonable inferences in favor of the plaintiff as required under Ninth Circuit precedent

The district court legally erred by not applying the "clear and convincing" evidence standard when it held the claims of the '446 Patent invalid. The district court also erred by not drawing all reasonable inferences from the alleged intrinsic evidence in favor of PurePredictive.

"A patent shall be presumed valid. Each claim of a patent . . . shall be presumed valid independently of the validity of other claims; dependent or multiple dependent claims shall be presumed valid even though dependent upon an invalid claim." 35 U.S.C. § 282. The burden of establishing patent invalidity rests on the party asserting invalidity. *Id.*

Challenges to patent validity under § 101 must satisfy a "clear and convincing" evidence standard. *See Microsoft*, 564 U.S. at 103 (holding that the presumption of validity under 35 U.S.C. § 282 requires the clear and convincing evidence standard to apply to all validity challenges because "we must presume that Congress intended to incorporate the heightened standard of proof, unless the statute otherwise dictates" and nothing "suggests that Congress meant to ... enact a standard of proof that would rise and fall with the facts of each case" (internal quotes omitted)); *CLS Bank*, 717 F.3d at 1304-05 *aff'd* 134 S. Ct. 2347 (2014); *DataTern*,

2015 WL 5190715 at *8–9 (applying the presumption of validity and the clear-and-convincing standard in § 101 analysis).

The district court's Order dismissed the Complaint on grounds that the patent claims are *invalid* under § 101. (Appx13:8-9) Yet the Order is devoid of any reference to or discussion of 35 U.S.C. § 282, the presumption of patent validity, or the burden to establish invalidity by "clear and convincing" evidence. *See Microsoft Corp*, 564 U.S. at 103. (Appx2-13) Nothing in the Order suggests that the district court's determination of invalidity took into account the presumption of patent validity by imposing the required "clear and convincing" evidence burden of proof on defendant. Indeed, under the heading "Legal Standard", the Order recites numerous cases regarding pleading standards and motions to dismiss, such as *Bell Atl. Corp. v. Twombly*, 550 U.S. 544, 570 (2007) and *Ashcroft v. Iqbal*, 556 U.S. 662, 678 (2009). (Appx4:13-20) The Order even cites *Genetic Techs. Ltd. v. Merial L.L.C.*, 818 F.3d 1269, 1373 (Fed. Cir. 2016) for the proposition that "it is possible . . . to determine patent eligibility under 35 U.S.C. § 101 on a Rule 12(b)(6) motion." (Appx5:1-4) But the Order is deafeningly silent about the "clear and convincing" evidence standard for proving patent invalidity. (Appx4-5)

To the contrary, the district court appeared to put the burden to prove validity squarely on the patentee, PurePredictive. For example, despite numerous examples that can be found throughout the specification and in the claims, the dismissal order

asserts that PurePredictive "fails to identify any previously existing technology that its claims improve upon, or that its claims do more than carry out regression analysis and evaluation." (Appx10:26-28) But no mention is made in the Order regarding defendant's burden, what evidence defendant proffered for invalidity, or whether or not defendant met it burden to demonstrate invalidity.

Defendant proffered no extrinsic evidence to support its assertion of invalidity. Indeed, defendant's arguments were based largely on "soundbites" from the specification that were taken out of context. The district court, rather than evaluate the claim language in view of the specification, adopted defendant's conclusory assertions about the claims. For example, defendant asserted that the claims were "directed to the manipulation of mathematical functions." (Appx126:8) Defendant argued that claim construction was unnecessary and proffered its overbroad abstraction in nigh conclusory fashion. (Appx124:22-26). And the district court accepted defendant's overbroad abstraction of the claims without considering the core features of the claims in view of the '446 specification.

Thus, the district court legally erred by finding the claims of the '446 Patent invalid without application of the "clear and convincing" evidence standard of proof.

Furthermore, a copy of the '446 Patent was included as part of the Complaint (*i.e.*, as part of the allegations). (Appx57-80) The Complaint alleges that the '446 Patent is valid and enforceable. (Appx48:3). Thus, intrinsic evidence of the meaning

of the asserted claims and supporting claim validity comprise part of the factual allegations set forth in the Complaint. Under Ninth Circuit precedent, the district court was required to "accept[] the plaintiff's allegations as true and draw[] all reasonable inferences in favor of the plaintiff." (Appx4:20-22). *See Usher*, 828 F.2d at 561 (9th Cir. 1987). Thus, the district court should have drawn all reasonable inferences from the intrinsic evidence, *e.g.*, the specification and claims, in favor of PurePredictive. Yet the Order did not meaningfully analyze, taking as true and in its best light favorable to PurePredictive, what the '446 specification teaches about the claims at issue and their benefits. The district court legally erred by finding the claims of the '446 Patent invalid without drawing all reasonable inferences from the intrinsic evidence in favor of PurePredictive.

Accordingly, the district court's finding that the claims of the '446 Patent are invalid should be reversed.

E. The district court erroneously held *all* claims invalid; amendment of the Complaint would not be futile

PurePredictive's Complaint alleges that defendant "infringes one or more claims" of the '446 Patent. (Appx48:10-11) Even though the Complaint identifies 1, 14, 17, and 23 as examples of claims that are infringed, PurePredictive's infringement action was not intended to be limited to the independent claims. But the dismissal order held that "[a]mendment would be futile in light of the analysis above." (Appx13:12)

In essence, the district court held all claims of the ‘446 Patent invalid without considering any dependent claim language covering core features of the claimed invention. This is contrary to the plain language of 35 U.S.C. § 282, which states that: “Each claim of a patent . . . shall be presumed valid independently of the validity of other claims; dependent or multiple dependent claims shall be presumed valid even though dependent upon an invalid claim.” 35 U.S.C. § 282.

There are numerous dependent claims claiming additional core features that defendant infringes. By way of example, dependent claim 12 (depends to claim 1) includes the following language:

> 12. The apparatus of claim 1, further comprising an orchestration module configured to direct workload data through the predictive ensemble based on the evaluation metadata data to produce a classification for the workload data and a confidence metric for the classification, the evaluation metadata synthesized to form the rule set for the subset of learned functions.

(Appx35, 24:7-13)

Notably, claim 12 not only covers the orchestration module for directing workload data through the structured environment of the predictive ensemble, but also includes the classification of data and providing a confidence metric for said classification. In other words, claim 12 is directed to transformation of data by the predictive ensemble, which is clearly patent eligible under § 101. *In re Bilski*, 545 F.3d at 963; *see Abele*, 684 F.2d 902 (upholding invention that takes X-ray attenuation data and displays it in visual form on a screen). Thus, even if amendment

to the Complaint were required to assert the dependent claims, such amendment would not be futile.

Accordingly, the dismissal order should be reversed.

CONCLUSION AND STATEMENT OF RELIEF SOUGHT

For the reasons stated herein, PurePredictive respectfully requests that this Court reverse the final judgment entered by district court and hold each of the claims-at-issue patent eligible under § 101.

Dated: January 12, 2018

Respectfully submitted,

By: /s/ Perry S. Clegg

Perry S. Clegg
R. Jeremy Adamson
KUNZLER, PC
50 W. Broadway, Suite 1000
Salt Lake City, Utah 84101
Tel: (801) 994- 4646
pclegg@kunzlerlaw.com

Counsel for Appellant PurePredictive, Inc.

CERTIFICATE OF FILING AND SERVICE

I hereby certify that, on January 12, 2018, I electronically filed the foregoing APPELLANT'S APPEAL BRIEF with the Clerk of Court using the CM/ECF System, which will send notice of such filing to all registered users.

I further certify that, upon acceptance and request from the Court, the required paper copies of the foregoing will be deposited with United Parcel Service for delivery to the Clerk, UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT, 717 Madison Place, N.W., Washington, D.C. 20439.

The necessary filing and service were performed in accordance with the instructions given to me by counsel in this case.

Dated: January 12, 2018

/s/ Priscilla C. Winkler

Priscilla C. Winkler
GibsonMoore Appellate Services, LLC
Post Office Box 1460
Richmond, VA 23218
(804) 249-7770
www.gibsonmoore.net

CERTIFICATE OF COMPLIANCE

1. This brief complies with the type-volume limitation of Fed. R. App. P. 32(a)(7)(B) because:

   this brief contains 13,976 words, excluding the parts of the brief exempted by Fed. R. App. P. 32(a)(7)(B)(iii).

2. This brief complies with the typeface requirements of Fed. R. App. P. 32(a)(5) and the type style requirements of Fed. R. App. P. 32(a)(6) because:

   this brief has been prepared in a proportionally spaced typeface using Microsoft Word in 14 point Times New Roman.

Dated: January 12, 2018

Respectfully submitted,

By: /s/ Perry S. Clegg

Perry S. Clegg
KUNZLER, PC
50 W. Broadway, Suite 1000
Salt Lake City, Utah 84101
Tel: (801) 994- 4646
pclegg@kunzlerlaw.com

Counsel for Appellant PurePredictive, Inc.

ADDENDUM

TABLE OF CONTENTS

Addendum

Page:

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

PUREPREDICTIVE, INC., Plaintiff, v. H2O.AI, INC., Defendant.	Case No. 17-cv-03049-WHO **JUDGMENT IN A CIVIL CASE** Re: Dkt. No. 31

Pursuant to the Court's Order Granting Hwo.AI, Inc.'s motion to dismiss, Judgment is accordingly entered in favor defendant of the and against plaintiff.

Dated: August 30, 2017

Susan Y. Soong, Clerk

Jean M. Davis

By: Jean M. Davis, Deputy Clerk

United States District Court
Northern District of California

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

PUREPREDICTIVE, INC., Plaintiff, v. H2O.AI, INC., Defendant.	Case No. 17-cv-03049-WHO **ORDER GRANTING H20.AI, INC.'S MOTION TO DISMISS** Re: Dkt. No. 14

INTRODUCTION

Plaintiff PUREPREDICTIVE, Inc. ("PPI") brings this action against H20.AI, Inc. ("H20") for direct and induced infringement of U.S. Patent No. 8,880,446 ("the '446 Patent"). H20 moves to dismiss PPI's claims, arguing that the '446 Patent is invalid because its claims are directed to patent-ineligible concepts under 35 U.S.C. § 101. Because PPI's claims are directed to the abstract concept of the manipulation of mathematical functions and make use of computers only as tools, rather than provide a specific improvement on a computer-related technology, I GRANT H20's motion to dismiss the Complaint.

BACKGROUND

I. Factual Background

A. The '446 Patent

The '446 Patent, titled "PREDICTIVE ANALYTICS FACTORY," relates to "an automated factory for predictive analytics." Complaint ("Compl.") Ex. A ("The '446 Patent") at 1:15–16 [Dkt. No. 1-1]. It describes that while "[d]ata analytics models are typically highly tuned and customized for a particular application" requiring "complex manual tools," such customized models are "typically useless or at least inaccurate for other applications and fields." *Id.* at 1:20–

32. On the other hand, "a general purpose analytics framework typically is not specialized enough for most applications without substantial customization." *Id.* at 1:32–34. There is thus a need for "an apparatus, system, method, and computer program product to generate a predictive ensemble in an automated manner . . . regardless of the particular field or application, with little or no input from a user or expert." *Id.* at 1:38–45. The '446 Patent purports to fill this gap.

Claim 14 is representative of the method, and recites the following elements:

> A method for a predictive analysis factory, the method comprising:
> pseudo-randomly generating a plurality of learned functions based on training data without prior knowledge regarding suitability of the generated learned functions for the training data, the training data received for forming a predictive ensemble customized for the training data;
> evaluating the plurality of learned functions using test data to generate evaluation metadata indicating an effectiveness of different learned functions at making predictions based on different subsets of test data; and
> forming the predictive ensemble comprising a subset of multiple learned functions from the plurality of learned functions, the subset of multiple learned functions selected and combined based on the evaluation metadata, the predictive ensemble comprising a rule set synthesized from the evaluation metadata to direct different subsets of the workload data through different learned functions of the multiple learned functions based on the evaluation metadata.

Compl. ¶ 17. Put very simply, the method performs predictive analytics in three steps. First, it receives data and generates "learned functions," or, for example, regressions from that data. *See* '446 Patent at 8:66–9:12. Second, it evaluates the effectiveness of those learned functions at making accurate predictions based on the test data. Finally, it selects the most effective learned functions and creates a rule set for additional data input. These three steps comprise the predictive analytics factory's method. Claim 1 recites a module-based apparatus for this predictive analytics factory, Compl. ¶ 16, Claim 17 recites a computer program product to perform the operations of the predictive analytics factory, Compl. ¶ 18, and Claim 23 recites a predictive analytics ensemble, Compl. ¶ 19.

B. The Parties and Procedural Background

PPI is a technology company that uses artificial intelligence to provide insight into business's data through the use of predictive modeling. Compl. ¶¶ 7–8. It is the owner of the '446 Patent. *Id.* ¶ 11. H20 is an open-source software company that provides a machine learning

United States District Court
Northern District of California

platform called "H20 with AutoML," integrated with applications and data products. *Id.* ¶¶ 9, 15. PPI alleges that H20 with AutoML infringes on Claims 1, 14, 17, and 23 of the '446 Patent. *Id.* ¶ 20. H20's website, which includes links to source code documentation, tutorials, videos, examples, and presentations about the platform, reinforces PPI's belief that the platform infringes on its claims. *Id.* ¶¶ 21–36. On May 24, 2017, PPI informed H20 of the '446 Patent and PPI's belief that H20's machine learning platform uses one or more apparatuses, methods, program products, and systems covered by the patent. *Id.* ¶ 14.

PPI filed suit against H20 on May 26, 2017, alleging both direct infringement of the '446 Patent as well as induced infringement of the '446 Patent through the H20 with AutoML platform. H20 now moves to dismiss PPI's Complaint.

LEGAL STANDARD

Under Federal Rule of Procedure 12(b)(6), a district court must dismiss a complaint if it fails to state a claim upon which relief can be granted. To survive a Rule 12(b)(6) motion to dismiss, the plaintiff must allege "enough facts to state a claim to relief that is plausible on its face." *Bell Atl. Corp. v. Twombly*, 550 U.S. 544, 570 (2007). A claim is facially plausible when the plaintiff pleads facts that "allow[] the court to draw the reasonable inference that the defendant is liable for the misconduct alleged." *Ashcroft v. Iqbal*, 556 U.S. 662, 678 (2009) (citation omitted). While courts do not require "heightened fact pleading of specifics," a plaintiff must allege facts sufficient to "raise a right to relief above the speculative level." *Twombly*, 550 U.S. at 555, 570. In deciding whether the plaintiff has stated a claim upon which relief can be granted, the court accepts the plaintiff's allegations as true and draws all reasonable inferences in favor of the plaintiff. *See Usher v. City of Los Angeles*, 828 F.2d 556, 561 (9th Cir. 1987). The court is not required to accept as true "allegations that are merely conclusory, unwarranted deductions of fact, or unreasonable inferences." *In re Gilead Scis. Sec. Litig.*, 536 F.3d 1049, 1055 (9th Cir. 2008).

To state a claim for patent infringement, "a patentee need only plead facts sufficient to place the alleged infringer on notice. This requirement ensures that the accused infringer has sufficient knowledge of the facts alleged to enable it to answer the complaint and defend itself."

United States District Court
Northern District of California

Phonometrics, Inc. v. Hosp. Franchise Sys., Inc., 203 F.3d 790, 794 (Fed. Cir. 2000). The Federal Circuit has "repeatedly recognized that in many cases it is possible and proper to determine patent eligibility under 35 U.S.C. § 101 on a Rule 12(b)(6) motion." *Genetic Techs. Ltd. v. Merial L.L.C.*, 818 F.3d 1269, 1373 (Fed. Cir. 2016). In such circumstances where it is possible and proper, "claim construction is not an inviolable prerequisite to a validity determination under § 101." *Bancorp Servs., L.L.C. v. Sun Life Assurance Co. of Can.*, 687 F.3d 1266, 1273 (Fed. Cir. 2012).

DISCUSSION

Under Section 101 of the Patent Act, "[w]hoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor . . . ." 35 U.S.C. § 101. The Supreme Court "has long held that this provision contains an important implicit exception: Laws of nature, natural phenomena, and abstract ideas are not patentable." *Alice Corp. Pty. Ltd. v. CLS Bank Int'l*, 134 S. Ct. 2347, 2354 (2014). The reason for the exception is clear enough—"such discoveries are manifestations of . . . nature, free to all men and reserved exclusively to none." *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, 132 S. Ct. 1289, 1293 (2012) (internal quotation marks and citations omitted). The boundaries of the exception, however, are not so clear.

The *Alice* court highlighted "the concern that drives this exclusionary principle as one of pre-emption." *Alice*, 134 S. Ct. at 2354 (noting the delicate balance inherent in promoting progress, the primary object of patent law, and granting a monopoly, the means for accomplishing that goal). In other words, patents that seek to wholly preempt others from using a law of nature or an abstract idea—"the basic tools of scientific and technological work"—are invalid. *Id.* "Accordingly, in applying the § 101 exception, we must distinguish between patents that claim the buildin[g] block[s] of human ingenuity and those that integrate the building blocks into something more, thereby transform[ing] them into a patent-eligible invention." *Id.* (internal quotation marks and citations omitted).

In evaluating whether claims are patent eligible, I must first "determine whether the claims at issue are directed to one of those patent-ineligible concepts." *Alice*, 134 S. Ct. at 2355. "[T]he

'directed to' inquiry applies a stage-one filter to claims, considered in light of the specification, based on whether their character as a whole is directed to excluded subject matter." *Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327, 1335 (Fed. Cir. 2016) (internal quotation marks omitted). Although there is no brightline rule for determining whether a claim is directed to an abstract idea, courts have articulated some guiding principles. When evaluating computer-related claims, courts look to whether the claims "improve the functioning of the computer itself," *Alice*, 134 S. Ct. at 2359, or whether "computers are invoked merely as a tool" to implement an abstract process. *Enfish*, 822 F.3d at 1336.

If the claims are directed to a patent-ineligible concept, I must then "consider the elements of each claim both individually and as an ordered combination to determine whether the additional elements transform the nature of the claim into a patent-eligible application." *Id.* at 1334 (internal quotation marks and citations omitted). This step entails the "search for an inventive concept—*i.e.*, an element or combination of elements that is sufficient to ensure that the patent in practice amounts to significantly more than a patent upon the [ineligible concept] itself." *Alice*, 134 S. Ct. at 2355 (internal quotation marks and citations omitted). "For the role of a computer in a computer-implemented invention to be deemed meaningful in the context of this analysis, it must involve more than performance of well-understood, routine, [and] conventional activities previously known to the industry." *Content Extraction & Transmission LLC v. Wells Fargo Bank, N.A.*, 776 F.3d 1343, 1347–48 (Fed. Cir. 2014). "[T]he mere recitation of a generic computer cannot transform a patent-ineligible abstract idea into a patent-eligible invention." *Id.* at 1348. However, "an inventive concept can be found in the non-conventional and non-generic arrangement of known, conventional pieces." *BASCOM Glob. Internet Servs., Inc. v. AT&T Mobility LLC*, 827 F.3d 1341, 1350 (Fed. Cir. 2016).

I. Whether the Claims Are Directed to Patent-Ineligible Concepts

H20 asserts that PPI's claims are directed to patent-ineligible concepts because they are directed to an abstract mathematical process for testing and refining algorithms, citing several cases in support of its assertion. They characterize PPI's patent as an attempt to monopolize the use of basic mathematical manipulations without reference to any specific implementation,

United States District Court
Northern District of California

application, purpose, or use. PPI counters that its claims are not only directed to computer-related technology but also solve a specific problem in and make improvements to computer-related technology. It too offers various cases it claims support its position.

While the Federal Circuit has recognized "that it is not always easy to determine the boundary between abstraction and patent-eligible subject matter," several of its cases have offered guiding principles. *Internet Patents Corp. v. Active Network, Inc.*, 790 F.3d 1343, 1347 (Fed. Cir. 2015); *see also Parker v. Flook*, 437 U.S. 584, 589 (1978) ("The line between a patentable 'process' and an unpatentable 'principle' is not always clear."). H20 urges that this case is similar to *Synopsys, Inc. v. Mentor Graphics Corp.*, in which the Federal Circuit addressed whether certain asserted claims were directed to a mental process or to a computer-related technology. 839 F.3d 1138, 1146–51 (Fed. Cir. 2016). The patent at issue related to the logic circuit design process and provided a scheme to translate functional descriptions of logic circuits into hardware component descriptions, without the requirement of certain intermediary steps, through constructs known as control flow graphs and assignment conditions. *Id.* at 1139–40. Explaining that "[t]he § 101 inquiry must focus on the language of the Asserted Claims themselves," the court noted that while "the inventions . . . were intended to be used in conjunction with computer-based designed tools, the [claims] were not confined to that conception" and "[we]re devoid of any reference to a computer or any other physical component." *Id.* at 1147, 1149. The claims, on their face, "d[id] not call for any form of computer implementation of the claimed methods," and the patent holder did not argue that the claims "must be *construed* as requiring a computer to perform the recited steps." *Id.* at 1149.

PPI distinguishes its patent specification and claims from those in *Synopsys*, pointing to various references to computers found in both. Instead, it suggests that I analyze the case under *Enfish*, where the patents at issue were directed to "an innovative logical model for a computer database" with a self-referential property, as opposed to the standard relational model. 822 F.3d at 1330. The Federal Circuit disagreed with the district court that the claims were directed to the abstract idea of "the concept of organizing information using tabular formats." *Id.* at 1337. It explained that "the claims [we]re not simply directed to *any* form of storing tabular data, but

instead are specifically directed to a *self-referential* table for a computer database." *Id.* The self-referential table was "an improvement of an existing technology" in that it offered "increased flexibility, faster search times, and smaller memory requirements." *Id.* The court contrasted this claim to others that simply "recited use of an abstract mathematical formula on any general purpose computer." *Id.* at 1338.

PPI also points to *McRO, Inc. v. Bandai Namco Games America, Inc.*, where the patents related to automating the process of 3-D animation of a character as it speaks. 837 F.3d 1299, 1303 (Fed. Cir. 2016). In examining whether the patent was directed to an abstract idea, the court explained that the computer "perform[e]d a distinct process to automate a task previously performed by humans" but went "beyond merely organizing [existing] information into a new form or carrying out a fundamental economic process." *Id.* at 1314–15 (internal quotation marks omitted). Instead, it "use[d] a combined order of specific rules that renders information into a specific format that is then used and applied to create desired results: a sequence of synchronized, animated characters." *Id.* at 1315. Thus, the court concluded that the claim was not directed to an abstract idea. *Id.* at 1316.

FairWarning IP, LLC v. Iatric Systems, Inc. is illustrative of a claim that the Federal Circuit did not consider an improvement of an existing technological process. 839 F.3d 1089 (Fed. Cir. 2016). It involved a patent on an invention that "collect[ed] information regarding accesses of a patient's personal health information, analyze[d] the information according to one of several rules (i.e., related to accesses in excess of a specific volume, accesses during a pre-determined time interval, or accesses by a specific user) to determine if the activity indicate[d] improper access, and provide[d] notification if it determines that improper access has occurred." *Id.* at 1093. The court explained that "analyzing information by steps people go through in their minds, or by mathematical algorithms, without more," are "essentially mental processes within the abstract-idea category." *Id.* Similarly, "merely presenting the results of abstract processes of collecting and analyzing information, without more (such as identifying a particular tool for presentation), is abstract as an ancillary part of such collection and analysis." *Id.* Applied to that patent, the court concluded that the claims were "directed to a

combination of these abstract-idea categories," specifically, "collecting and analyzing information to detect misuse and notifying a user when misuse is detected." *Id.* at 1094. Although the claims "us[ed] one of a few possible rules to analyze the audit log data," those rules were nonetheless directed to an abstract idea. *Id.* The mere "use of the computer," rather than "the incorporation of the claimed rules," was not enough to "improve [the] existing technological process." *Id.* The Federal Circuit affirmed the district court's dismissal of the suit on the pleadings. *Id.* at 1097.

Turning to this case, I agree with H20 that PPI's claims are directed to a mental process and the abstract concept of using mathematical algorithms to perform predictive analytics. The method of the predictive analytics factory is directed towards collecting and analyzing information. The first step, generating learned functions or regressions from data—the basic mathematical process of, for example, regression modeling, or running data through an algorithm—is not a patentable concept. *See DDR Holdings, LLC v. Hotels.com, L.P.*, 773 F.3d 1245, 1256 (Fed. Cir. 2014) ("We know that mathematical algorithms, including those executed on a generic computer, are abstract ideas."). That the "function generator module" described in the '446 Patent "may generate hundreds, thousands, or millions of learned functions, or more," '446 Patent at 9:55–57, does not change this conclusion. While PPI claims that this shows it would be impossible for a human to perform such a task, just because a computer can make calculations more quickly than a human does not render a method patent eligible. *See Bancorp Servs., L.L.C. v. Sun Life Assurance Co. of Canada (U.S.)*, 687 F.3d 1266, 1278 (Fed. Cir. 2012) ("The computer required by some of Bancorp's claims is employed only for its most basic function, the performance of repetitive calculations, and as such does not impose meaningful limits on the scope of those claims."); *Synopsys*, 839 F.3d at 1146 ("Methods which can be performed entirely in the human mind are unpatentable not because there is anything wrong with claiming mental method steps as part of a process containing non-mental steps, but rather because [they] embody the 'basic tools of scientific and technological work' that are free to all men and reserved exclusively to none."). The patent specification's description of this process as a "brute force, trial-and-error approach," reinforces that this process is merely the running of data through a machine. '446 Patent, 10:6–10.

The next steps of the method are similarly abstract. The method takes the learned functions, evaluates their effectiveness, and selects those most effective to create a rule set. These are mathematical processes that not only could be performed by humans but also go to the general abstract concept of predictive analytics rather than any specific application. Again, the patent specification's language reinforces this conclusion; it describes "an apparatus, system, method, and computer program product [that] would comprise a predictive analytics factory configured to generate a predictive ensemble *regardless of the particular field or application*." '446 Patent at 1:41–44 (emphasis added). These claims are similar to those in *Synopsys;* that the claims "were intended to be used in conjunction with computer-based design tools" did not save the broad and abstract language of the asserted claims. *Synopsys*, 839 F.3d at 1149.

If I accept PPI's assertion that the claims are directed to a computer-related technology, PPI still cannot show that its claims improve the functioning of a computer-related technology rather than use computers as a tool. *See Alice*, 134 S. Ct. at 2359; *Enfish*, 822 F.3d at 1336. While PPI points to Claim 17 (which specifically references a "computer program product"), other claims' use of the term "module," and the patent specification's reference to the learned functions comprising "a computer readable code," '446 Patent at 8:50–51, these few passing references to computers only show that the method uses the computer as a tool for automation of its process. *See DDR Holdings*, 773 F.3d at 1256 ("[R]ecitation of generic computer limitations does not make an otherwise ineligible claim patent-eligible. The bare fact that a computer exists in the physical rather than purely conceptual realm is beside the point." (internal quotation marks and citations omitted)).

In *Enfish*, for example, the claims improved upon the standard relational model with a new type of self-referential database, which constituted more than "recited use of an abstract mathematical formula on any general purpose computer." 822 F.3d at 1338. And in *McRo*, the technology went beyond mere "organiz[ation] [of existing] information into a new form or carrying out a fundamental economic process." 837 F.3d at 1314–15. But PPI fails to identify any previously existing technology that its claims improve upon, or that its claims do more than carry out regression analysis and evaluation. Instead, its claims "merely present[] the results of

abstract processes of collecting and analyzing information" and "us[e] one of a few possible rules to analyze the [] data." *FairWarning*, 839 F.3d at 1094.

PPI's claims are directed to the patent-ineligible abstract concept of testing and refining mathematical algorithms. While they may invoke computers as a tool for this process, the claims do not make a specific improvement on an existing computer-related technology. Because PPI's claims are directed to patent-ineligible concepts, I will move on to *Alice*'s step two.

II. Whether the Additional Elements Transform the Nature of the Claim into a Patent-Eligible Application

H20 argues that PPI's claims do not contain an inventive concept sufficient to transform them into a patent-eligible application. PPI counters that its claims contain both an unconventional improvement in its field and an inventive concept through its ordered combination. It likens its claims to those in *DDR Holdings* and *BASCOM*. It describes its claims as aiming to "generate a predictive ensemble in an automated manner" with "little or no input from a user or expert," while still offering customization and finely tuned predictive ensembles. Opp. at 19–20. PPI points out that its ensembles "do not need extensive tuning and customization" and "are applicable regardless of the particular field or application." Opp. at 20 (internal quotation marks omitted). It alleges that this process constitutes an unconventional solution in its field and that its particular arrangement is an improvement to existing technology.

In *DDR Holdings*, the Federal Circuit examined a patent that claimed a technical solution to the "Internet-centric problem" of third-party web merchants luring the host website's visitor traffic away when visitors would click on merchants' advertisements. 773 F.3d at 1248, 1259. The patents disclosed a system that created a new composite webpage displaying the product information from the third-party merchant, but retaining the host website's "look and feel" and allowing the host website to retain its visitor traffic. *Id.* at 1248–49. The court found that the "new, hybrid webpage that merges content" from two sources and creates a "store within a store" was an inventive solution to the problem of customer loss tied specifically to the Internet. *Id.* at 1257–58. The court explicitly "caution[ed], however, that not all claims purporting to address Internet-centric challenges are eligible for patent." *Id.* at 1258.

United States District Court
Northern District of California

It is easy to distinguish the present case. For starters, PPI does not claim that its patent is Internet-centric, but instead that its claims are "necessarily rooted in computer technology" and "take place in a technological environment." Opp. at 18. The realm of computer technology or technological environments is far broader than Internet-centric challenges. Moreover, while the claims in *DDR Holdings* were directed to a very specific problem—that allowing third-party advertising on websites resulted in decreased visitor retention—here, PPI's claims address the universal problem in any analytical framework of choosing between a more generally applicable or more specific and customized model. And finally, while the solutions in *DDR Holdings* were specifically engineered to construct a hybrid web page "stor[ing] visually perceptible elements from the identified host website," 773 F.3d at 1257 (internal quotation marks omitted), PPI's solutions remain the abstract mathematical processes of collecting and analyzing data. This is not the unconventional or inventive solution necessary to satisfy the second step of *Alice*.

Nor are PPI's claims like those in *BASCOM*. In that case, the claims at issue were directed to "a content filtering system for filtering content retrieved from an Internet computer network." 827 F.3d at 1348 (internal quotation marks omitted). The filtering system was "located on a remote ISP server that . . . allow[ed] individual network accounts to customize the filtering of Internet traffic associated with the account." *Id.* at 1345. While the Federal Circuit found that the claims were directed to the abstract idea of content filtration, they agreed with the patent holder that they contained the inventive concept of "installation of a filtering tool at a specific location, remote from end-users, with customizable filter features specific to each end user." *Id.* at 1350. But the court again clarified that the "claims d[id] not merely recite the abstract idea of filtering content along with the requirement to perform it on the Internet, or to perform it on a set of generic computer components," because "[s]uch claims would not contain an inventive concept." *Id.*

PPI's claims cannot meet that showing. While the claims in *BASCOM* were specifically tied not only to the Internet but also to the specific function of content filtration as well as concrete locations, PPI's claims "merely recite the abstract idea of" predictive analytics "along with the requirement to perform it on . . . a set of generic computer components." *BASCOM*, 827 F.3d at 1350. PPI's claims do not describe specific system architecture, and references to generic

"modules" do not provide any further specificity. While PPI claims that "the ordered combination of the claims" provides an inventive concept, there is nothing inventive about its particular arrangement. Instead, its claims recite the functional steps for collecting, analyzing, and refining data through mathematical algorithms. As *BASCOM* explained, an inventive concept "must be significantly more than the abstract idea itself." *Id.* at 1349. PPI's technology, while perhaps an effective method, is simply an implementation of the basic concept of predictive analytics on an apparatus, computer program product, or other medium.

Because PPI cannot show an inventive concept sufficient to transform its claims, its claims fail as patent ineligible under Section 101.

CONCLUSION

For the reasons stated above, I GRANT H20's motion to dismiss the Complaint. Amendment would be futile in light of the analysis above; leave to amend was not sought and will not be granted. Judgment shall be entered accordingly.

IT IS SO ORDERED.

Dated: August 29, 2017

William H. Orrick
United States District Judge

US008880446B2

(12) **United States Patent**
Wellman et al.

(10) **Patent No.: US 8,880,446 B2**
(45) **Date of Patent: Nov. 4, 2014**

(54) **PREDICTIVE ANALYTICS FACTORY**

(71) Applicant: **CloudVu, Inc.**, Sandy, UT (US)

(72) Inventors: **Richard W. Wellman**, Park City, UT (US); **Kelly D. Phillipps**, Salt Lake City, UT (US)

(73) Assignee: **PurePredictive, Inc.**, Sandy, UT (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **13/870,861**

(22) Filed: **Apr. 25, 2013**

(65) **Prior Publication Data**

US 2014/0136452 A1 May 15, 2014

Related U.S. Application Data

(60) Provisional application No. 61/727,114, filed on Nov. 15, 2012.

(51) **Int. Cl.**
G06N 99/00 (2010.01)

(52) **U.S. Cl.**
CPC ***G06N 99/005*** (2013.01)
USPC **706/12**

(58) **Field of Classification Search**
USPC 706/12
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,112,304 A	8/2000	Clawson	
6,507,726 B1	1/2003	Atkinson et al.	
6,523,015 B1	2/2003	Bera et al.	
7,219,085 B2	5/2007	Buck et al.	
7,689,520 B2	3/2010	Burges et al.	
7,814,194 B2	10/2010	Hellerstein et al.	
8,160,981 B2	4/2012	Aparicio	
8,209,271 B1 *	6/2012	Lin et al.	706/12
8,209,274 B1 *	6/2012	Lin et al.	706/21
8,214,308 B2 *	7/2012	Chu	706/14
8,229,864 B1 *	7/2012	Lin et al.	706/11
8,250,009 B1 *	8/2012	Breckenridge et al.	706/14
8,311,967 B1 *	11/2012	Lin et al.	706/45
8,370,279 B1 *	2/2013	Lin et al.	706/12
8,370,280 B1 *	2/2013	Lin et al.	706/12
8,527,324 B2 *	9/2013	Richter	705/7.38
2004/0059966 A1	3/2004	Chan et al.	

(Continued)

FOREIGN PATENT DOCUMENTS

DE	102010046439	3/2012
WO	2012151198	11/2012

OTHER PUBLICATIONS

J. Zico Kolter, et al., "Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts", Journal of Machine Learning Research 8 (2007), pp. 2755-2790.*

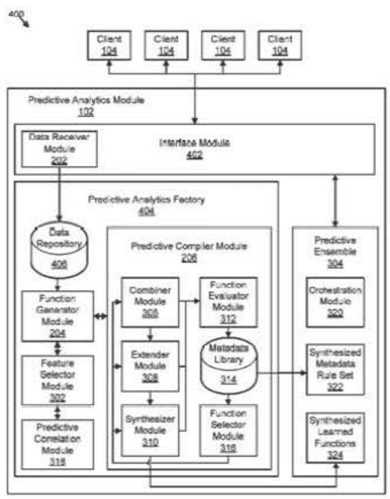
(Continued)

Primary Examiner — Kakalo Chaki
Assistant Examiner — Fuming Wu
(74) *Attorney, Agent, or Firm* — Kunzler Law Group, PC

(57) **ABSTRACT**

An apparatus, system, method, and computer program product are disclosed for a predictive analytics factory. A receiver module is configured to receive training data. A function generator module is configured to determine a plurality of learned functions from multiple classes based on the training data. A predictive compiler module is configured to form a predictive ensemble comprising a subset of learned functions from the plurality of learned functions. The subset of learned functions is from at least two of the multiple classes.

25 Claims, 8 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

2005/0132052	A1	6/2005	Uttamchandani et al.	
2005/0228789	A1	10/2005	Fawcett et al.	
2007/0111179	A1	5/2007	Hochwarth et al.	
2008/0162487	A1*	7/2008	Richter	707/10
2009/0035733	A1	2/2009	Meitar et al.	
2009/0186329	A1	7/2009	Connor	
2009/0254379	A1*	10/2009	Adams et al.	705/4
2009/0327172	A1	12/2009	Liu et al.	
2010/0009330	A1	1/2010	Yaskin	
2010/0114663	A1*	5/2010	Casas et al.	705/10
2010/0131314	A1	5/2010	Ting et al.	
2010/0223212	A1*	9/2010	Manolescu et al.	706/12
2010/0306141	A1	12/2010	Chidlovskii	
2011/0111384	A1	5/2011	Dietrich et al.	
2011/0119300	A1	5/2011	Marcade	
2011/0137672	A1*	6/2011	Adams et al.	705/2
2012/0004893	A1*	1/2012	Vaidyanathan et al.	703/11
2012/0030160	A1*	2/2012	Ratnam et al.	706/47
2012/0158620	A1	6/2012	Paquet et al.	
2012/0158624	A1*	6/2012	Lingenfelder et al.	706/12
2012/0191630	A1*	7/2012	Breckenridge et al.	706/12
2012/0191631	A1*	7/2012	Breckenridge et al.	706/12
2012/0284212	A1*	11/2012	Lin et al.	706/12
2012/0284213	A1*	11/2012	Lin et al.	706/12
2012/0284600	A1*	11/2012	Lin et al.	715/212
2013/0004930	A1	1/2013	Sorenson et al.	
2014/0136452	A1*	5/2014	Wellman et al.	706/12

OTHER PUBLICATIONS

Bennett, "EHRs connect research and practice:Where predictive modeling,artificial intelligence,and clinical decision support intersect," Health Policy and Technology (2012) 1, 105-114; Available online Apr. 18, 2012.*

Kolter, "DynamicWeighted Majority: An Ensemble Method for Drifting Concepts," Journal of Machine Learning Research 8 (2007) 2755-2790.*

McTigue, Jake, "Predictive Analytics for IT", InformationWeek Reports, Mar. 2012, pp. 17, Reports. InformationWeek.com.

Purpura, Stephen, "Statistically Adaptive Learning for a General Class of Cost Functions (SA L-BFGS)", Sep. 5, 2012, pp. 7, arXiv:1209.0029v3 [cs.LG] Sep. 5, 2012.

Application No. PCT/US2013/077236, 3128.2.2pct, International Search Report and Written Opinion, Feb. 24, 2014.

Application No. PCT/US2013/070358, 3128.2.1pct, International Search Report and Written Opinion, Feb. 28, 2014.

Application No. PCT/US2014/013028, 3128.2.3pct, International Search Report and Written Opinion, May 15, 2014.

U.S. Appl. No. 13/749,618, Office Action, Jul. 8, 2014.

Application No. PCT/US14/10729, International Search Report and Written Opinion, Aug. 28, 2014.

* cited by examiner

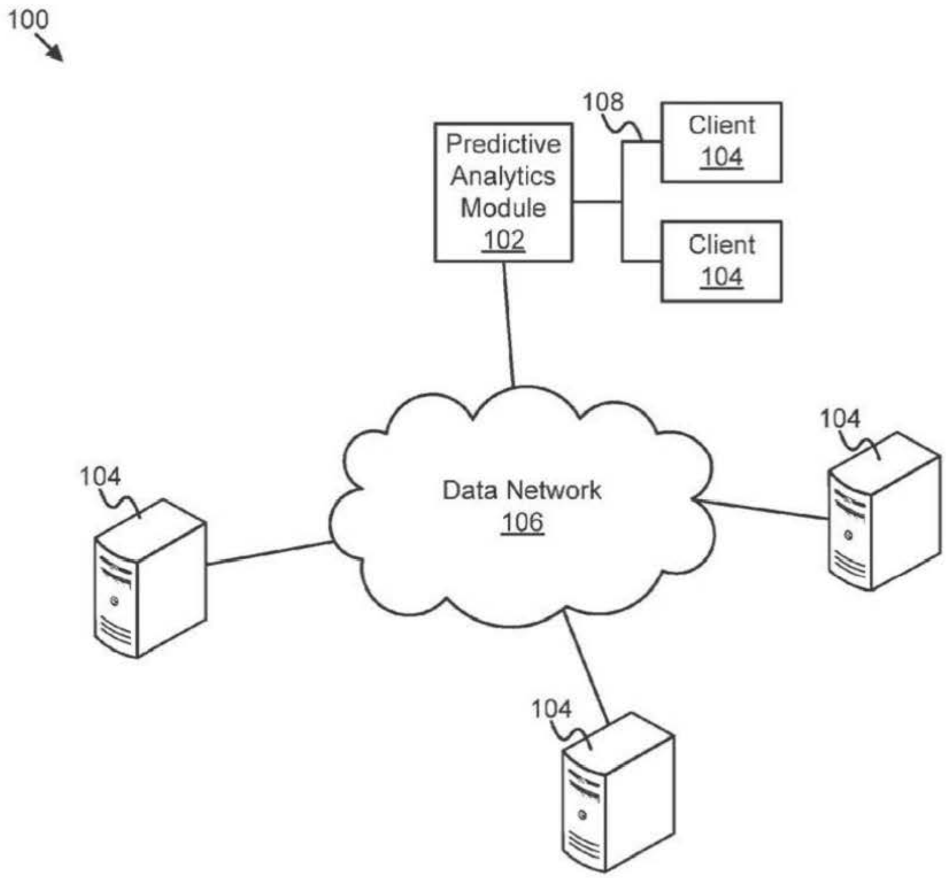

100
108
Predictive
Analytics
Module
102
Client
104
Client
104
Data Network
106
104
104
104

FIG. 1

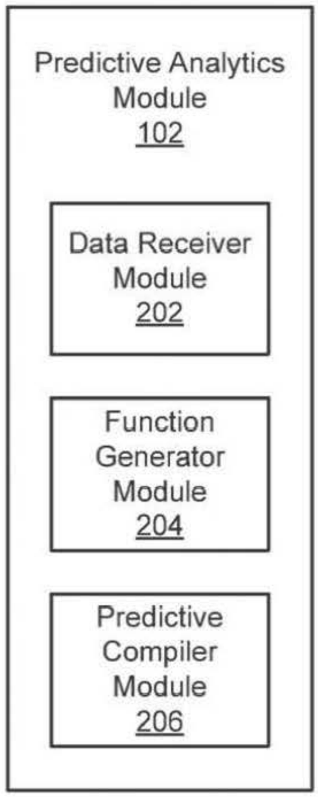

Predictive Analytics Module 102
Data Receiver Module 202
Function Generator Module 204
Predictive Compiler Module 206

FIG. 2

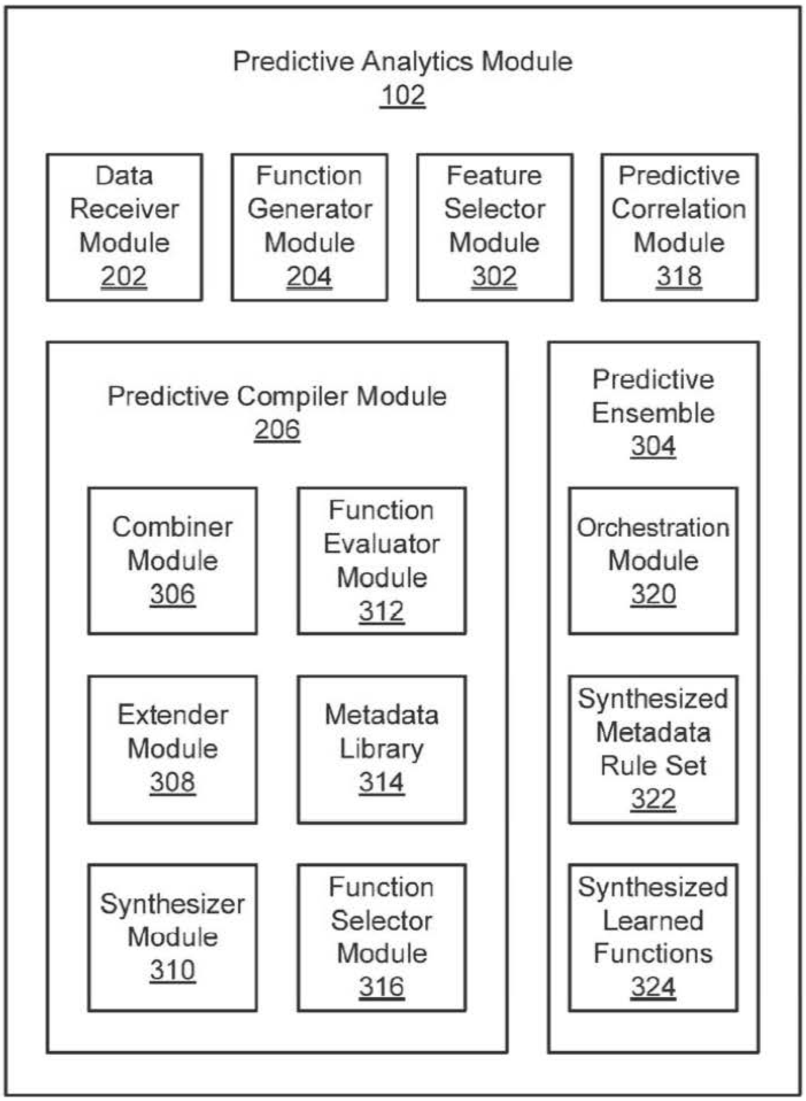

Predictive Analytics Module
102
Data Receiver Module 202
Function Generator Module 204
Feature Selector Module 302
Predictive Correlation Module 318
Predictive Compiler Module 206
Combiner Module 306
Function Evaluator Module 312
Extender Module 308
Metadata Library 314
Synthesizer Module 310
Function Selector Module 316
Predictive Ensemble 304
Orchestration Module 320
Synthesized Metadata Rule Set 322
Synthesized Learned Functions 324

FIG. 3

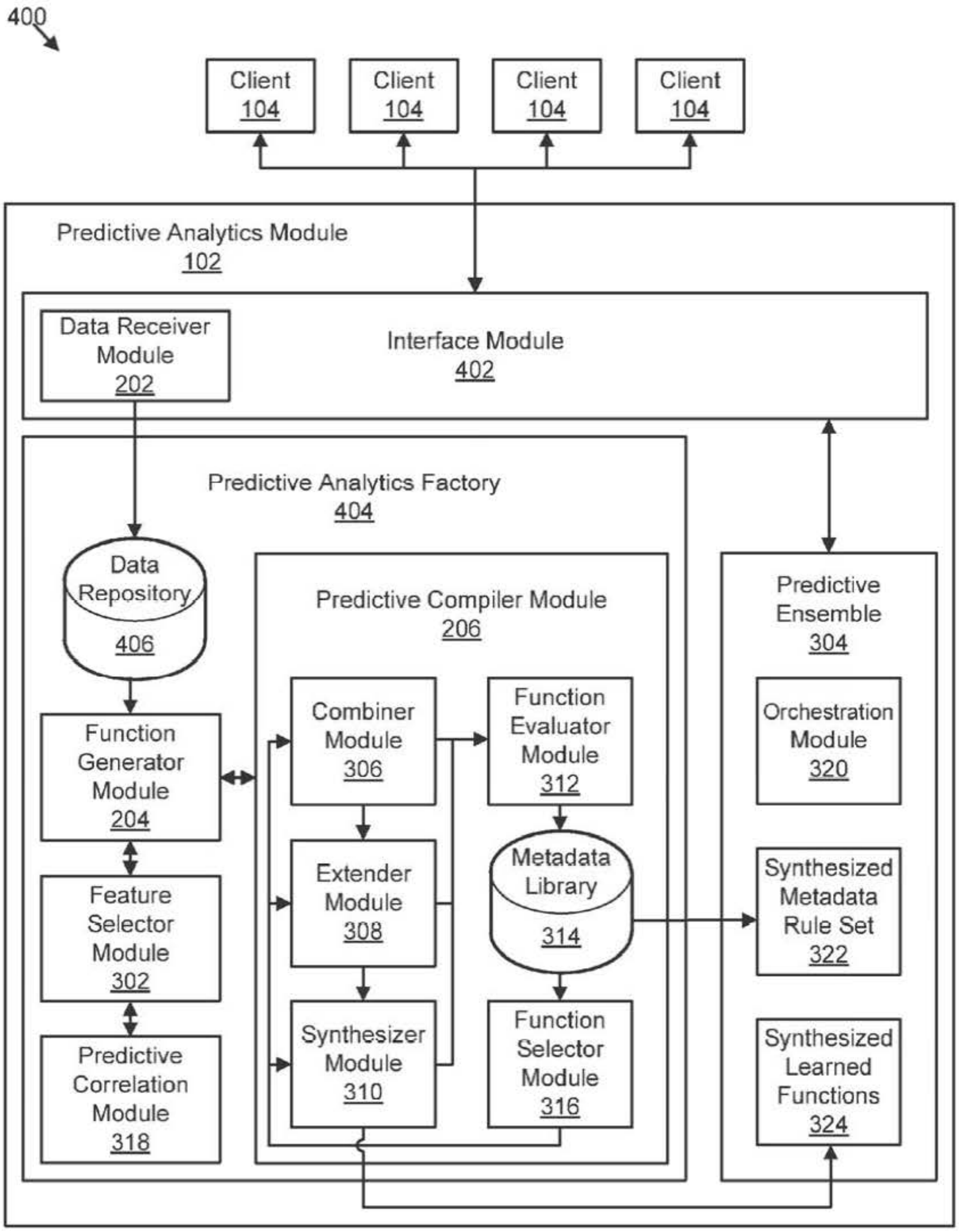

400
Client
104
Client
104
Client
104
Client
104
Predictive Analytics Module
102
Data Receiver
Module
202
Interface Module
402
Predictive Analytics Factory
404
Data
Repository
406
Predictive Compiler Module
206
Predictive
Ensemble
304
Function
Generator
Module
204
Combiner
Module
306
Function
Evaluator
Module
312
Orchestration
Module
320
Extender
Module
308
Metadata
Library
314
Feature
Selector
Module
302
Synthesized
Metadata
Rule Set
322
Synthesizer
Module
310
Function
Selector
Module
316
Predictive
Correlation
Module
318
Synthesized
Learned
Functions
324

FIG. 4

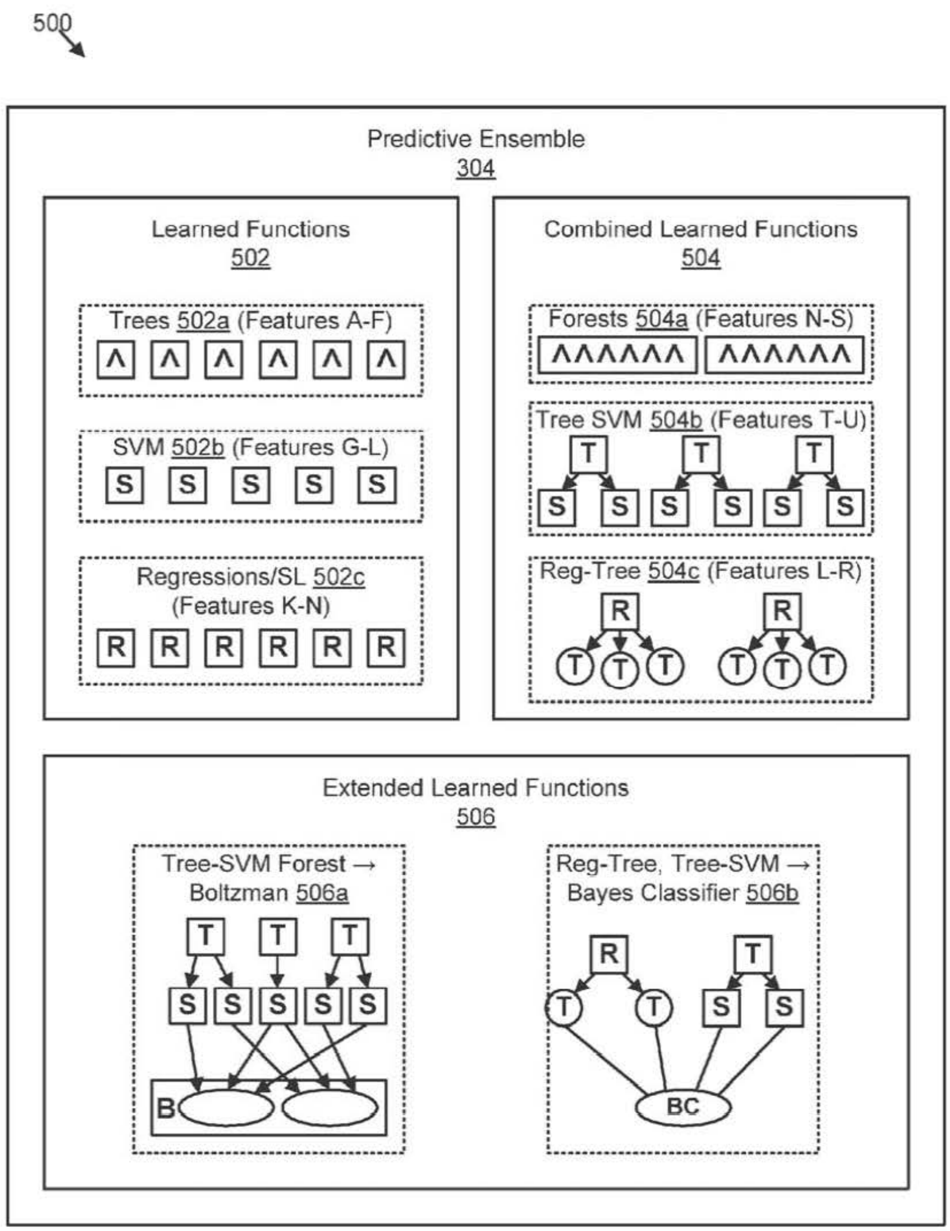

500
Predictive Ensemble
304
Learned Functions
502
Trees 502a (Features A-F)
SVM 502b (Features G-L)
Regressions/SL 502c
(Features K-N)
Combined Learned Functions
504
Forests 504a (Features N-S)
Tree SVM 504b (Features T-U)
Reg-Tree 504c (Features L-R)
Extended Learned Functions
506
Tree-SVM Forest →
Boltzman 506a
Reg-Tree, Tree-SVM →
Bayes Classifier 506b
BC

FIG. 5

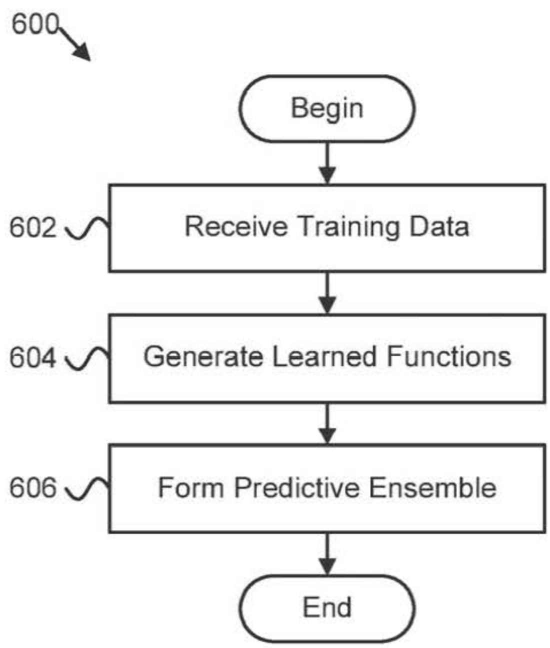

600
Begin
602
Receive Training Data
604
Generate Learned Functions
606
Form Predictive Ensemble
End

FIG. 6

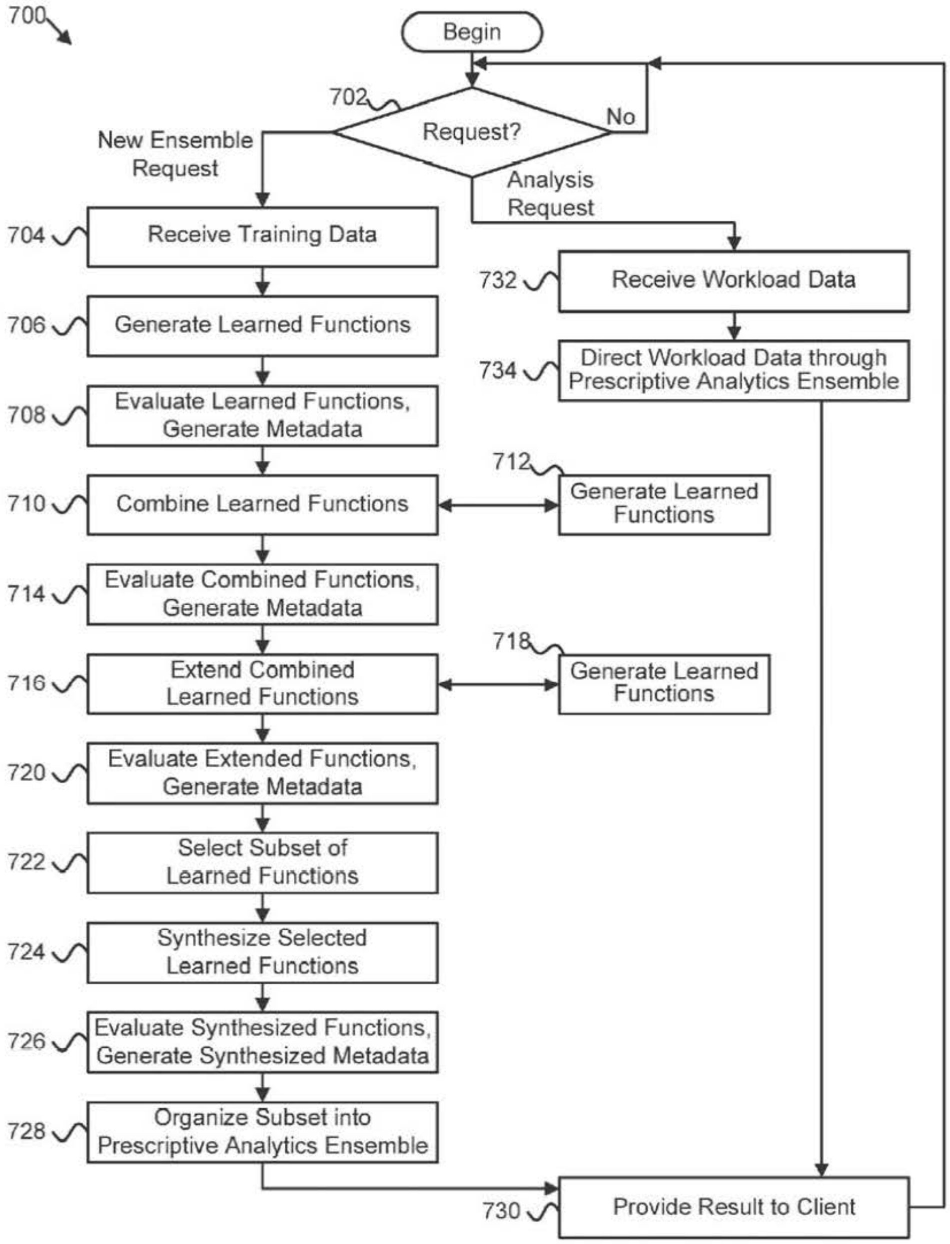

700
Begin
702
Request?
No
New Ensemble Request
Analysis Request
704
Receive Training Data
706
Generate Learned Functions
708
Evaluate Learned Functions, Generate Metadata
710
Combine Learned Functions
712
Generate Learned Functions
714
Evaluate Combined Functions, Generate Metadata
716
Extend Combined Learned Functions
718
Generate Learned Functions
720
Evaluate Extended Functions, Generate Metadata
722
Select Subset of Learned Functions
724
Synthesize Selected Learned Functions
726
Evaluate Synthesized Functions, Generate Synthesized Metadata
728
Organize Subset into Prescriptive Analytics Ensemble
732
Receive Workload Data
734
Direct Workload Data through Prescriptive Analytics Ensemble
730
Provide Result to Client

FIG. 7

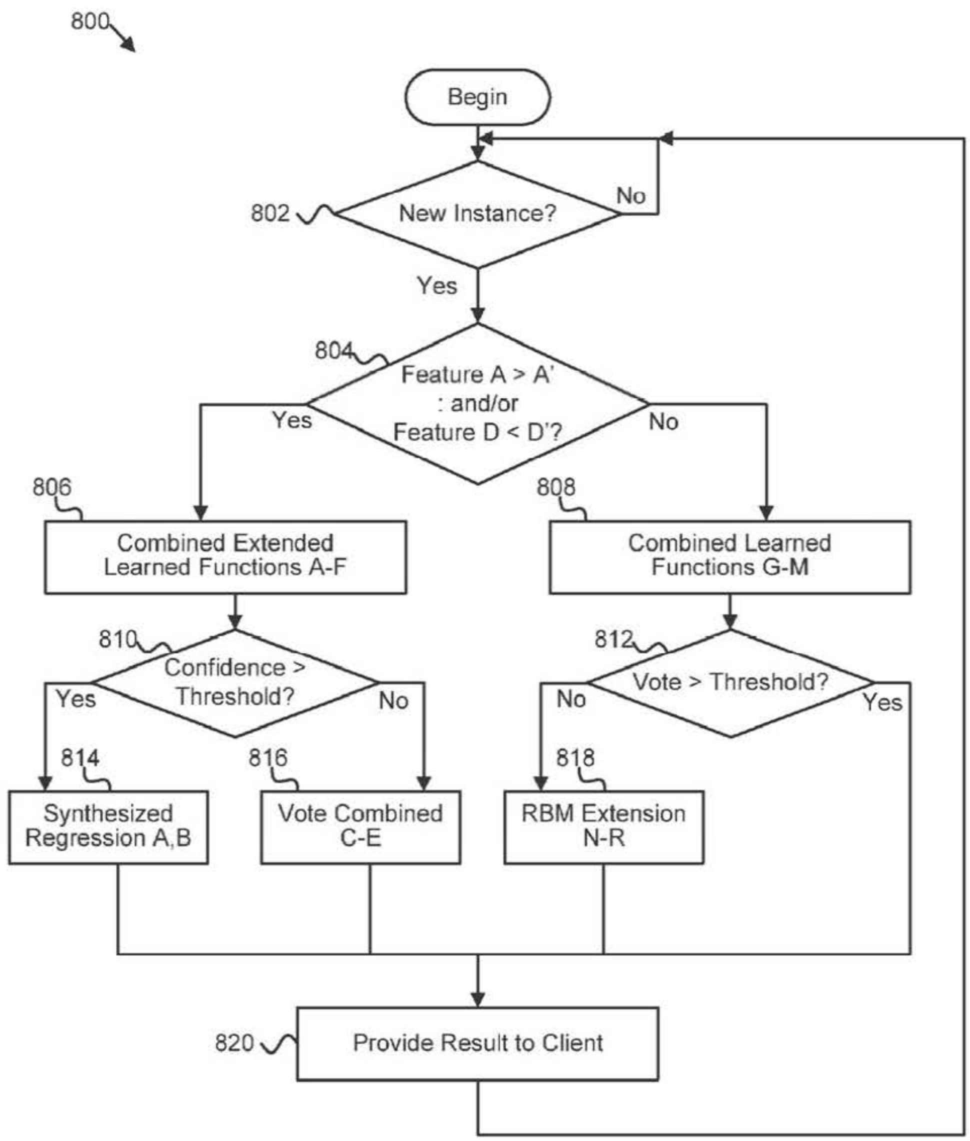

800
Begin
802
New Instance?
No
Yes
804
Feature A > A'
: and/or
Feature D < D'?
Yes
No
806
Combined Extended
Learned Functions A-F
808
Combined Learned
Functions G-M
810
Confidence >
Threshold?
Yes
No
812
Vote > Threshold?
No
Yes
814
Synthesized
Regression A,B
816
Vote Combined
C-E
818
RBM Extension
N-R
820
Provide Result to Client

FIG. 8

PREDICTIVE ANALYTICS FACTORY

CROSS-REFERENCES TO RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Patent Application No. 61/727,114 entitled "PREDICTIVE ANALYTICS FACTORY" and filed on Nov. 15, 2012 for Richard W. Wellman, et al., which is incorporated herein by reference.

TECHNICAL FIELD

The present disclosure, in various embodiments, relates to analytics and more particularly relates to an automated factory for predictive analytics.

BACKGROUND

Data analytics models are typically highly tuned and customized for a particular application. Such tuning and customization often requires pre-existing knowledge about the particular application, and can require the use of complex manual tools to achieve this tuning and customization. For example, an expert in a certain field may carefully tune and customize an analytics model for use in the expert's field using a manual tool.

While a highly tuned, expert customized analytics model may be useful for a particular application or field, because of the high level of tuning and customization, the analytics model is typically useless or at least inaccurate for other applications and fields. Conversely, a general purpose analytics framework typically is not specialized enough for most applications without substantial customization.

SUMMARY

From the foregoing discussion, it should be apparent that a need exists for an apparatus, system, method, and computer program product to generate a predictive ensemble in an automated manner. Beneficially, such an apparatus, system, method, and computer program product would comprise a predictive analytics factory configured to generate a predictive ensemble regardless of the particular field or application, with little or no input from a user or expert.

The present disclosure has been developed in response to the present state of the art, and in particular, in response to the problems and needs in the art that have not yet been fully solved by currently available analytics methods. Accordingly, the present disclosure has been developed to provide an apparatus, system, method, and computer program product for a predictive analytics factory that overcome many or all of the above-discussed shortcomings in the art.

Apparatuses are presented for a predictive analytics factory. In one embodiment, a receiver module is configured to receive training data. A function generator module, in certain embodiments, is configured to determine a plurality of learned functions from multiple classes based on the training data. A predictive compiler module, in a further embodiment, is configured to form a predictive ensemble comprising a subset of learned functions from the plurality of learned functions. In one embodiment, the subset of learned functions is from at least two of the multiple classes.

Methods are presented for a predictive analytics factory. In one embodiment, a method includes pseudo-randomly generating a plurality of learned functions based on training data without prior knowledge regarding suitability of the generated learned functions for the training data. A method, in another embodiment, includes evaluating a plurality of learned functions using test data to generate evaluation metadata. In another embodiment, a method includes forming a predictive ensemble comprising a subset of learned functions from a plurality of learned functions, where the subset of learned functions are selected based on evaluation metadata.

Computer program products are presented, comprising a computer readable storage medium storing computer usable program code executable to perform operations for a predictive analytics factory. In one embodiment, an operation includes determining a plurality of learned functions using training data comprising a plurality of features. An operation, in another embodiment, includes selecting a subset of features of training data based on evaluation metadata generated for a plurality of learned functions. In a further embodiment, an operation includes forming a predictive ensemble comprising at least two learned functions from a plurality of learned functions that use a selected subset of features.

A predictive analytics ensemble is presented. In one embodiment, a predictive analytics ensemble includes a plurality of learned functions synthesized from a larger plurality of learned functions. In a further embodiment, a predictive analytics ensemble includes a metadata rule set synthesized from evaluation metadata for a plurality of learned functions. A predictive analytics ensemble, in another embodiment, includes an orchestration module configured to direct data through a plurality of learned functions based on a synthesized metadata rule set to produce a result.

Reference throughout this specification to features, advantages, or similar language does not imply that all of the features and advantages that may be realized with the present disclosure should be or are in any single embodiment of the disclosure. Rather, language referring to the features and advantages is understood to mean that a specific feature, advantage, or characteristic described in connection with an embodiment is included in at least one embodiment of the present disclosure. Thus, discussion of the features and advantages, and similar language, throughout this specification may, but do not necessarily, refer to the same embodiment.

Furthermore, the described features, advantages, and characteristics of the disclosure may be combined in any suitable manner in one or more embodiments. The disclosure may be practiced without one or more of the specific features or advantages of a particular embodiment. In other instances, additional features and advantages may be recognized in certain embodiments that may not be present in all embodiments of the disclosure.

These features and advantages of the present disclosure will become more fully apparent from the following description and appended claims, or may be learned by the practice of the disclosure as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

In order that the advantages of the disclosure will be readily understood, a more particular description of the disclosure briefly described above will be rendered by reference to specific embodiments that are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the disclosure and are not therefore to be considered to be limiting of its scope, the disclosure will be described and explained with additional specificity and detail through the use of the accompanying drawings, in which:

FIG. **1** is a schematic block diagram illustrating one embodiment of a system for a predictive analytics factory;

FIG. **2** is a schematic block diagram illustrating one embodiment of a predictive analytics module;

FIG. **3** is a schematic block diagram illustrating another embodiment of a predictive analytics module;

FIG. **4** is a schematic block diagram illustrating another embodiment of a system for a predictive analytics factory;

FIG. **5** is a schematic block diagram illustrating one embodiment of learned functions for a predictive ensemble;

FIG. **6** is a schematic flow chart diagram illustrating one embodiment of a method for a predictive analytics factory;

FIG. **7** is a schematic flow chart diagram illustrating another embodiment of a method for a predictive analytics factory; and

FIG. **8** is a schematic flow chart diagram illustrating one embodiment of a method for directing data through a predictive ensemble.

DETAILED DESCRIPTION

Aspects of the present disclosure may be embodied as a system, method or computer program product. Accordingly, aspects of the present disclosure may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "circuit," "module" or "system." Furthermore, aspects of the present disclosure may take the form of a computer program product embodied in one or more computer readable storage media having computer readable program code embodied thereon.

Many of the functional units described in this specification have been labeled as modules, in order to more particularly emphasize their implementation independence. For example, a module may be implemented as a hardware circuit comprising custom VLSI circuits or gate arrays, off-the-shelf semiconductors such as logic chips, transistors, or other discrete components. A module may also be implemented in programmable hardware devices such as field programmable gate arrays, programmable array logic, programmable logic devices or the like.

Modules may also be implemented in software for execution by various types of processors. An identified module of executable code may, for instance, comprise one or more physical or logical blocks of computer instructions which may, for instance, be organized as an object, procedure, or function. Nevertheless, the executables of an identified module need not be physically located together, but may comprise disparate instructions stored in different locations which, when joined logically together, comprise the module and achieve the stated purpose for the module.

Indeed, a module of executable code may be a single instruction, or many instructions, and may even be distributed over several different code segments, among different programs, and across several memory devices. Similarly, operational data may be identified and illustrated herein within modules, and may be embodied in any suitable form and organized within any suitable type of data structure. The operational data may be collected as a single data set, or may be distributed over different locations including over different storage devices, and may exist, at least partially, merely as electronic signals on a system or network. Where a module or portions of a module are implemented in software, the software portions are stored on one or more computer readable storage media.

Any combination of one or more computer readable storage media may be utilized. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing.

More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), a portable compact disc read-only memory (CD-ROM), a digital versatile disc (DVD), a Blu-ray disc, an optical storage device, a magnetic tape, a Bernoulli drive, a magnetic disk, a magnetic storage device, a punch card, integrated circuits, other digital processing apparatus memory devices, or any suitable combination of the foregoing, but would not include propagating signals. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

Computer program code for carrying out operations for aspects of the present disclosure may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Python, C++ or the like and conventional procedural programming languages, such as the "C" programming language or similar programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

Reference throughout this specification to "one embodiment," "an embodiment," or similar language means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present disclosure. Thus, appearances of the phrases "in one embodiment," "in an embodiment," and similar language throughout this specification may, but do not necessarily, all refer to the same embodiment, but mean "one or more but not all embodiments" unless expressly specified otherwise. The terms "including," "comprising," "having," and variations thereof mean "including but not limited to" unless expressly specified otherwise. An enumerated listing of items does not imply that any or all of the items are mutually exclusive and/or mutually inclusive, unless expressly specified otherwise. The terms "a," "an," and "the" also refer to "one or more" unless expressly specified otherwise.

Furthermore, the described features, structures, or characteristics of the disclosure may be combined in any suitable manner in one or more embodiments. In the following description, numerous specific details are provided, such as examples of programming, software modules, user selections, network transactions, database queries, database structures, hardware modules, hardware circuits, hardware chips, etc., to provide a thorough understanding of embodiments of the disclosure. However, the disclosure may be practiced without one or more of the specific details, or with other methods, components, materials, and so forth. In other instances, well-known structures, materials, or operations are not shown or described in detail to avoid obscuring aspects of the disclosure.

Aspects of the present disclosure are described below with reference to schematic flowchart diagrams and/or schematic block diagrams of methods, apparatuses, systems, and computer program products according to embodiments of the disclosure. It will be understood that each block of the schematic flowchart diagrams and/or schematic block diagrams, and combinations of blocks in the schematic flowchart diagrams and/or schematic block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the schematic flowchart diagrams and/or schematic block diagrams block or blocks.

These computer program instructions may also be stored in a computer readable storage medium that can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions stored in the computer readable storage medium produce an article of manufacture including instructions which implement the function/act specified in the schematic flowchart diagrams and/or schematic block diagrams block or blocks.

The computer program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other devices to cause a series of operational steps to be performed on the computer, other programmable apparatus or other devices to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide processes for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

The schematic flowchart diagrams and/or schematic block diagrams in the Figures illustrate the architecture, functionality, and operation of possible implementations of apparatuses, systems, methods and computer program products according to various embodiments of the present disclosure. In this regard, each block in the schematic flowchart diagrams and/or schematic block diagrams may represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function(s).

It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. Other steps and methods may be conceived that are equivalent in function, logic, or effect to one or more blocks, or portions thereof, of the illustrated figures.

Although various arrow types and line types may be employed in the flowchart and/or block diagrams, they are understood not to limit the scope of the corresponding embodiments. Indeed, some arrows or other connectors may be used to indicate only the logical flow of the depicted embodiment. For instance, an arrow may indicate a waiting or monitoring period of unspecified duration between enumerated steps of the depicted embodiment. It will also be noted that each block of the block diagrams and/or flowchart diagrams, and combinations of blocks in the block diagrams and/or flowchart diagrams, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

The description of elements in each figure may refer to elements of proceeding figures. Like numbers refer to like elements in all figures, including alternate embodiments of like elements.

FIG. **1** depicts one embodiment of a system **100** for a predictive analytics factory. The system **100**, in the depicted embodiment, includes a predictive analytics module **102** that is in communication with several clients **104** over a data network **106**, and with several clients **104** over a local channel **108**, such as a system bus, an application programming interface (API), or the like. A client **104** may comprise a software application, a user, a hardware computing device with a processor and memory, or another entity in communication with the predictive analytics module **102**. In general, the predictive analytics module **102** generates predictive ensembles for the clients **104**. In certain embodiments, the predictive analytics module **102** provides a predictive analytics framework allowing clients **104** to request predictive ensembles, to make analysis requests, and to receive predictive results, such as a classification, a confidence metric, an inferred function, a regression function, an answer, a prediction, a recognized pattern, a rule, a recommendation, or other results.

Predictive analytics is the study of past performance, or patterns, found in historical and transactional data to identify behavior and trends in future events. This may be accomplished using a variety of statistical techniques including modeling, machine learning, data mining, or the like.

One term for large, complex, historical data sets is Big Data. Examples of Big Data include web logs, social networks, blogs, system log files, call logs, customer data, user feedback, or the like. These data sets may often be so large and complex that they are awkward and difficult to work with using traditional tools. With technological advances in computing resources, including memory, storage, and computational power, along with frameworks and programming models for data-intensive distributed applications, the ability to collect, analyze and mine these huge repositories of structured, unstructured, and/or semi-structured data has only recently become possible.

In certain embodiments, prediction may be applied through at least two general techniques: Regression and Classification.

Regression models attempt to fit a mathematical equation to approximate the relationship between the variables being analyzed. These models may include "Discrete Choice" models such as Logistic Regression, Multinomial Logistic Regression, Probit Regression, or the like. When factoring in time, Time Series models may be used, such as Auto Regression—AR, Moving Average—MA, ARMA, AR Conditional Heteroskedasticity—ARCH, Generalized ARCH—GARCH and Vector AR—VAR). Other models include Survival or Duration analysis, Classification and Regression Trees (CART), Multivariate Adaptive Regression Splines (MARS), and the like.

Classification is a form of artificial intelligence that uses computational power to execute complex algorithms in an effort to emulate human cognition. One underlying problem, however, remains: determining the set of all possible behaviors given all possible inputs is much too large to be included in a set of observed examples. Classification methods may include Neural Networks, Radial Basis Functions, Support Vector Machines, Naïve Bayes, k-Nearest Neighbors, Geospatial Predictive modeling, and the like.

Each of these forms of modeling make assumptions about the data set and model the given data, however, some models

are more accurate than others and none of the models are ideal. Historically, using predictive analytics tools was a cumbersome and difficult process, often involving the engagement of a Data Scientist or other expert. Any easier-to-use tools or interfaces for general business users, however, typically fall short in that they still require "heavy lifting" by IT personnel in order to present and massage data and results. A Data Scientist typically must determine the optimal class of learning machines that would be the most applicable for a given data set, and rigorously test the selected hypothesis by first fine-tuning the learning machine parameters and second by evaluating results fed by trained data.

The predictive analytics module **102**, in certain embodiments, generates predictive ensembles for the clients **104**, with little or no input from a Data Scientist or other expert, by generating a large number of learned functions from multiple different classes, evaluating, combining, and/or extending the learned functions, synthesizing selected learned functions, and organizing the synthesized learned functions into a predictive ensemble. The predictive analytics module **102**, in one embodiment, services analysis requests for the clients **104** using the generated predictive ensembles.

By generating a large number of learned functions, without regard to the effectiveness of the generated learned functions, without prior knowledge of the generated learned functions suitability, or the like, and evaluating the generated learned functions, in certain embodiments, the predictive analytics module **102** may provide predictive ensembles that are customized and finely tuned for data from a specific client **104**, without excessive intervention or fine-tuning. The predictive analytics module **102**, in a further embodiment, may generate and evaluate a large number of learned functions using parallel computing on multiple processors, such as a massively parallel processing (MPP) system or the like.

The predictive analytics module **102** may service predictive analytics requests to clients **104** locally, executing on the same host computing device as the predictive analytics module **102**, by providing an API to clients **104**, receiving function calls from clients **104**, providing a hardware command interface to clients **104**, or otherwise providing a local channel **108** to clients **104**. In a further embodiment, the predictive analytics module **102** may service predictive analytics requests to clients **104** over a data network **106**, such as a local area network (LAN), a wide area network (WAN) such as the Internet as a cloud service, a wireless network, a wired network, or another data network **106**.

FIG. **2** depicts one embodiment of a predictive analytics module **102**. The predictive analytics module **102** of FIG. **2**, in certain embodiments, may be substantially similar to the predictive analytics module **102** described above with regard to FIG. **1**. In the depicted embodiment, the predictive analytics module **102** includes a data receiver module **202**, a function generator module **204**, and a predictive compiler module **206**.

The data receiver module **202**, in certain embodiments, is configured to receive client data, such as training data, test data, workload data, or the like, from a client **104**, either directly or indirectly. The data receiver module **202**, in various embodiments, may receive data over a local channel **108** such as an API, a shared library, a hardware command interface, or the like; over a data network **106** such as wired or wireless LAN, WAN, the Internet, a serial connection, a parallel connection, or the like. In certain embodiments, the data receiver module **202** may receive data indirectly from a client **104** through an intermediate module that may pre-process, reformat, or otherwise prepare the data for the predictive analysis module **102**. The data receiver module **202** may support structured data, unstructured data, semi-structured data, or the like.

One type of data that the data receiver module **202** may receive, as part of a new ensemble request or the like, is initialization data. The predictive analytics module **102**, in certain embodiments, may use initialization data to train and test learned functions from which the predictive analytics module **102** may build a predictive ensemble. Initialization data may comprise historical data, statistics, Big Data, customer data, marketing data, computer system logs, computer application logs, data networking logs, or other data that a client **104** provides to the data receiver module **202** with which to build, initialize, train, and/or test a predictive ensemble.

Another type of data that the data receiver module **202** may receive, as part of an analysis request or the like, is workload data. The predictive analytics module **102**, in certain embodiments, may process workload data using a predictive ensemble to obtain a result, such as a classification, a confidence metric, an inferred function, a regression function, an answer, a prediction, a recognized pattern, a rule, a recommendation, or the like. Workload data for a specific predictive ensemble, in one embodiment, has substantially the same format as the initialization data used to train and/or evaluate the predictive ensemble. For example, initialization data and/or workload data may include one or more features. As used herein, a feature may comprise a column, category, data type, attribute, characteristic, label, or other grouping of data. For example, in embodiments where initialization data and/or workload data that is organized in a table format, a column of data may be a feature. Initialization data and/or workload data may include one or more instances of the associated features. In a table format, where columns of data are associated with features, a row of data is an instance.

As described below with regard to FIG. **4**, in one embodiment, the data receiver module **202** may maintain client data, such as initialization data and/or workload data, in a data repository **406**, where the function generator module **204**, the predictive compiler module **206**, or the like may access the data. In certain embodiments, as described below, the function generator module **204** and/or the predictive compiler module **206** may divide initialization data into subsets, using certain subsets of data as training data for generating and training learned functions and using certain subsets of data as test data for evaluating generated learned functions.

The function generator module **204**, in certain embodiments, is configured to generate a plurality of learned functions based on training data from the data receiver module **202**. A learned function, as used herein, comprises a computer readable code that accepts an input and provides a result. A learned function may comprise a compiled code, a script, text, a data structure, a file, a function, or the like. In certain embodiments, a learned function may accept instances of one or more features as input, and provide a result, such as a classification, a confidence metric, an inferred function, a regression function, an answer, a prediction, a recognized pattern, a rule, a recommendation, or the like. In another embodiment, certain learned functions may accept instances of one or more features as input, and provide a subset of the instances, a subset of the one or more features, or the like as an output. In a further embodiment, certain learned functions may receive the output or result of one or more other learned functions as input, such as a Bayes classifier, a Boltzmann machine, or the like.

The function generator module **204** may generate learned functions from multiple different predictive analytics classes,

models, or algorithms. For example, the function generator module **204** may generate decision trees; decision forests; kernel classifiers and regression machines with a plurality of reproducing kernels; non-kernel regression and classification machines such as logistic, CART, multi-layer neural nets with various topologies; Bayesian-type classifiers such as Naïve Bayes and Boltzmann machines; logistic regression; multinomial logistic regression; probit regression; AR; MA; ARMA; ARCH; GARCH; VAR; survival or duration analysis; MARS; radial basis functions; support vector machines; k-nearest neighbors; geospatial predictive modeling; and/or other classes of learned functions.

In one embodiment, the function generator module **204** generates learned functions pseudo-randomly, without regard to the effectiveness of the generated learned functions, without prior knowledge regarding the suitability of the generated learned functions for the associated training data, or the like. For example, the function generator module **204** may generate a total number of learned functions that is large enough that at least a subset of the generated learned functions are statistically likely to be effective. As used herein, pseudo-randomly indicates that the function generator module **204** is configured to generate learned functions in an automated manner, without input or selection of learned functions, predictive analytics classes or models for the learned functions, or the like by a Data Scientist, expert, or other user.

The function generator module **204**, in certain embodiments, generates as many learned functions as possible for a requested predictive ensemble, given one or more parameters or limitations. A client **104** may provide a parameter or limitation for learned function generation as part of a new ensemble request or the like to an interface module **402** as described below with regard to FIG. **4**, such as an amount of time; an allocation of system resources such as a number of processor nodes or cores, or an amount of volatile memory; a number of learned functions; runtime constraints on the requested ensemble such as an indicator of whether or not the requested ensemble should provide results in real-time; and/or another parameter or limitation from a client **104**.

The number of learned functions that the function generator module **204** may generate for building a predictive ensemble may also be limited by capabilities of the system **100**, such as a number of available processors or processor cores, a current load on the system **100**, a price of remote processing resources over the data network **106**; or other hardware capabilities of the system **100** available to the function generator module **204**. The function generator module **204** may balance the hardware capabilities of the system **100** with an amount of time available for generating learned functions and building a predictive ensemble to determine how many learned functions to generate for the predictive ensemble.

In one embodiment, the function generator module **204** may generate at least 50 learned functions for a predictive ensemble. In a further embodiment, the function generator module **204** may generate hundreds, thousands, or millions of learned functions, or more, for a predictive ensemble. By generating an unusually large number of learned functions from different classes without regard to the suitability or effectiveness of the generated learned functions for training data, in certain embodiments, the function generator module **204** ensures that at least a subset of the generated learned functions, either individually or in combination, are useful, suitable, and/or effective for the training data without careful curation and fine tuning by a Data Scientist or other expert.

Similarly, by generating learned functions from different predictive analytics classes without regard to the effectiveness or the suitability of the different predictive analytics classes for training data, the function generator module **204**, in certain embodiments, may generate learned functions that are useful, suitable, and/or effective for the training data due to the sheer amount of learned functions generated from the different predictive analytics classes. This brute force, trial-and-error approach to generating learned functions, in certain embodiments, eliminates or minimizes the role of a Data Scientist or other expert in generation of a predictive ensemble.

The function generator module **204**, in certain embodiments, divides initialization data from the data receiver module **202** into various subsets of training data, and may use different training data subsets, different combinations of multiple training data subsets, or the like to generate different learned functions. The function generator module **204** may divide the initialization data into training data subsets by feature, by instance, or both. For example, a training data subset may comprise a subset of features of initialization data, a subset of features of initialization data, a subset of both features and instances of initialization data, or the like. Varying the features and/or instances used to train different learned functions, in certain embodiments, may further increase the likelihood that at least a subset of the generated learned functions are useful, suitable, and/or effective. In a further embodiment, the function generator module **204** ensures that the available initialization data is not used in its entirety as training data for any one learned function, so that at least a portion of the initialization data is available for each learned function as test data, which is described in greater detail below with regard to the function evaluator module **312** of FIG. **3**.

In one embodiment, the function generator module **204** may also generate additional learned functions in cooperation with the predictive compiler module **206**. The function generator module **204** may provide a learned function request interface, allowing the predictive compiler module **206** or another module, a client **104**, or the like to send a learned function request to the function generator module **204** requesting that the function generator module **204** generate one or more additional learned functions. In one embodiment, a learned function request may include one or more attributes for the requested one or more learned functions. For example, a learned function request, in various embodiments, may include a predictive analytics class for a requested learned function, one or more features for a requested learned function, instances from initialization data to use as training data for a requested learned function, runtime constraints on a requested learned function, or the like. In another embodiment, a learned function request may identify initialization data, training data, or the like for one or more requested learned functions and the function generator module **204** may generate the one or more learned functions pseudo-randomly, as described above, based on the identified data.

The predictive compiler module **206**, in one embodiment, is configured to form a predictive ensemble using learned functions from the function generator module **204**. As used herein, a predictive ensemble comprises an organized set of a plurality of learned functions. Providing a classification, a confidence metric, an inferred function, a regression function, an answer, a prediction, a recognized pattern, a rule, a recommendation, or another result using a predictive ensemble, in certain embodiments, may be more accurate than using a single learned function.

The predictive compiler module **206** is described in greater detail below with regard to FIG. **3**. The predictive compiler module **206**, in certain embodiments, may combine and/or

extend learned functions to form new learned functions, may request additional learned functions from the function generator module **204**, or the like for inclusion in a predictive ensemble. In one embodiment, the predictive compiler module **206** evaluates learned functions from the function generator module **204** using test data to generate evaluation metadata. The predictive compiler module **206**, in a further embodiment, may evaluate combined learned functions, extended learned functions, combined-extended learned functions, additional learned functions, or the like using test data to generate evaluation metadata.

The predictive compiler module **206**, in certain embodiments, maintains evaluation metadata in a metadata library **314**, as described below with regard to FIGS. **3** and **4**. The predictive compiler module **206** may select learned functions (e.g. learned functions from the function generator module **204**, combined learned functions, extended learned functions, learned functions from different predictive analytics classes, and/or combined-extended learned functions) for inclusion in a predictive ensemble based on the evaluation metadata. In a further embodiment, the predictive compiler module **206** may synthesize the selected learned functions into a final, synthesized function or function set for a predictive ensemble based on evaluation metadata. The predictive compiler module **206**, in another embodiment, may include synthesized evaluation metadata in a predictive ensemble for directing data through the predictive ensemble or the like.

FIG. **3** depicts another embodiment of a predictive analytics module **102**. The predictive analytics module **102** of FIG. **3**, in certain embodiments, may be substantially similar to the predictive analytics module **102** described above with regard to FIGS. **1** and **2**. In the depicted embodiment, the predictive analytics module **102** includes the data receiver module **202**, the function generator module **204**, and the predictive compiler module **206** described above with regard to FIG. **2** and further includes a feature selector module **302** a predictive correlation module **318**, and a predictive ensemble **304**. The predictive compiler module **206**, in the depicted embodiment, includes a combiner module **306**, an extender module **308**, a synthesizer module **310**, a function evaluator module **312**, a metadata library **314**, and a function selector module **316**. The predictive ensemble **304**, in the depicted embodiment, includes an orchestration module **320**, a synthesized metadata rule set **322**, and synthesized learned functions **324**.

In one embodiment, the feature selector module **302** determines which features of initialization data to use in the predictive ensemble **304**, and in the associated learned functions, and/or which features of the initialization data to exclude from the predictive ensemble **304**, and from the associated learned functions. As described above, initialization data, and the training data and test data derived from the initialization data, may include one or more features. Learned functions and the predictive ensembles **304** that they form are configured to receive and process instances of one or more features. Certain features may be more predictive than others, and the more features that the predictive compiler module **206** processes and includes in the generated predictive ensemble **304**, the more processing overhead used by the predictive compiler module **206**, and the more complex the generated predictive ensemble **304** becomes. Additionally, certain features may not contribute to the effectiveness or accuracy of the results from a predictive ensemble **304**, but may simply add noise to the results.

The feature selector module **302**, in one embodiment, cooperates with the function generator module **204** and the predictive compiler module **206** to evaluate the effectiveness of various features, based on evaluation metadata from the metadata library **314** described below. For example, the function generator module **204** may generate a plurality of learned functions for various combinations of features, and the predictive compiler module **206** may evaluate the learned functions and generate evaluation metadata. Based on the evaluation metadata, the feature selector module **302** may select a subset of features that are most accurate or effective, and the predictive compiler module **206** may use learned functions that utilize the selected features to build the predictive ensemble **304**. The feature selector module **302** may select features for use in the predictive ensemble **304** based on evaluation metadata for learned functions from the function generator module **204**, combined learned functions from the combiner module **306**, extended learned functions from the extender module **308**, combined extended functions, synthesized learned functions from the synthesizer module **310**, or the like.

In a further embodiment, the feature selector module **302** may cooperate with the predictive compiler module **206** to build a plurality of different predictive ensembles **304** for the same initialization data or training data, each different predictive ensemble **304** utilizing different features of the initialization data or training data. The predictive compiler module **206** may evaluate each different predictive ensemble **304**, using the function evaluator module **312** described below, and the feature selector module **302** may select the predictive ensemble **304** and the associated features which are most accurate or effective based on the evaluation metadata for the different predictive ensembles **304**. In certain embodiments, the predictive compiler module **206** may generate tens, hundreds, thousands, millions, or more different predictive ensembles **304** so that the feature selector module **302** may select an optimal set of features (e.g. the most accurate, most effective, or the like) with little or no input from a Data Scientist, expert, or other user in the selection process.

In one embodiment, the predictive compiler module **206** may generate a predictive ensemble **304** for each possible combination of features from which the feature selector module **302** may select. In a further embodiment, the predictive compiler module **206** may begin generating predictive ensembles **304** with a minimal number of features, and may iteratively increase the number of features used to generate predictive ensembles **304** until an increase in effectiveness or usefulness of the results of the generated predictive ensembles **304** fails to satisfy a feature effectiveness threshold. By increasing the number of features until the increases stop being effective, in certain embodiments, the predictive compiler module **206** may determine a minimum effective set of features for use in a predictive ensemble **304**, so that generation and use of the predictive ensemble **304** is both effective and efficient. The feature effectiveness threshold may be predetermined or hard coded, may be selected by a client **104** as part of a new ensemble request or the like, may be based on one or more parameters or limitations, or the like.

During the iterative process, in certain embodiments, once the feature selector module **302** determines that a feature is merely introducing noise, the predictive compiler module **206** excludes the feature from future iterations, and from the predictive ensemble **304**. In one embodiment, a client **104** may identify one or more features as required for the predictive ensemble **304**, in a new ensemble request or the like. The feature selector module **302** may include the required features in the predictive ensemble **304**, and select one or more of the remaining optional features for inclusion in the predictive ensemble **304** with the required features.

In a further embodiment, based on evaluation metadata from the metadata library **314**, the feature selector module

302 determines which features from initialization data and/or training data are adding noise, are not predictive, are the least effective, or the like, and excludes the features from the predictive ensemble **304**. In other embodiments, the feature selector module **302** may determine which features enhance the quality of results, increase effectiveness, or the like, and selects the features for the predictive ensemble **304**.

In one embodiment, the feature selector module **302** causes the predictive compiler module **206** to repeat generating, combining, extending, and/or evaluating learned functions while iterating through permutations of feature sets. At each iteration, the function evaluator module **312** may determine an overall effectiveness of the learned functions in aggregate for the current iteration's selected combination of features. Once the feature selector module **302** identifies a feature as noise introducing, the feature selector module may exclude the noisy feature and the predictive compiler module **206** may generate a predictive ensemble **304** without the excluded feature. In one embodiment, the predictive correlation module **318** determines one or more features, instances of features, or the like that correlate with higher confidence metrics (e.g., that are most effective in predicting results with high confidence). The predictive correlation module **318** may cooperate with, be integrated with, or otherwise work in concert with the feature selector module **302** to determine one or more features, instances of features, or the like that correlate with higher confidence metrics. For example, as the feature selector module **302** causes the predictive compiler module **206** to generate and evaluate learned functions with different sets of features, the predictive correlation module **318** may determine which features and/or instances of features correlate with higher confidence metrics, are most effective, or the like based on metadata from the metadata library **314**.

The predictive correlation module **318**, in certain embodiments, is configured to harvest metadata regarding which features correlate to higher confidence metrics, to determine which feature was predictive of which outcome or result, or the like. In one embodiment, the predictive correlation module **318** determines the relationship of a feature's predictive qualities for a specific outcome or result based on each instance of a particular feature. In other embodiments, the predictive correlation module **318** may determine the relationship of a feature's predictive qualities based on a subset of instances of a particular feature. For example, the predictive correlation module **318** may discover a correlation between one or more features and the confidence metric of a predicted result by attempting different combinations of features and subsets of instances within an individual feature's dataset, and measuring an overall impact on predictive quality, accuracy, confidence, or the like. The predictive correlation module **318** may determine predictive features at various granularities, such as per feature, per subset of features, per instance, or the like.

In one embodiment, the predictive correlation module **318** determines one or more features with a greatest contribution to a predicted result or confidence metric as the predictive compiler module **206** forms the predictive ensemble **304**, based on evaluation metadata from the metadata library **314**, or the like. For example, the predictive compiler module **206** may build one or more synthesized learned functions **324** that are configured to provide one or more features with a greatest contribution as part of a result. In another embodiment, the predictive correlation module **318** may determine one or more features with a greatest contribution to a predicted result or confidence metric dynamically at runtime as the predictive ensemble **304** determines the predicted result or confidence metric. In such embodiments, the predictive correlation module **318** may be part of, integrated with, or in communication with the predictive ensemble **304**. The predictive correlation module **318** may cooperate with the predictive ensemble **304**, such that the predictive ensemble **304** provides a listing of one or more features that provided a greatest contribution to a predicted result or confidence metric as part of a response to an analysis request.

In determining features that are predictive, or that have a greatest contribution to a predicted result or confidence metric, the predictive correlation module **318** may balance a frequency of the contribution of a feature and/or an impact of the contribution of the feature. For example, a certain feature or set of features may contribute to the predicted result or confidence metric frequently, for each instance or the like, but have a low impact. Another feature or set of features may contribute relatively infrequently, but has a very high impact on the predicted result or confidence metric (e.g. provides at or near 100% confidence or the like). While the predictive correlation module **318** is described herein as determining features that are predictive or that have a greatest contribution, in other embodiments, the predictive correlation module **318** may determine one or more specific instances of a feature that are predictive, have a greatest contribution to a predicted result or confidence metric, or the like.

In the depicted embodiment, the predictive compiler module **206** includes a combiner module **306**. The combiner module **306** combines learned functions, forming sets, strings, groups, trees, or clusters of combined learned functions. In certain embodiments, the combiner module **306** combines learned functions into a prescribed order, and different orders of learned functions may have different inputs, produce different results, or the like. The combiner module **306** may combine learned functions in different combinations. For example, the combiner module **306** may combine certain learned functions horizontally or in parallel, joined at the inputs and at the outputs or the like, and may combine certain learned functions vertically or in series, feeding the output of one learned function into the input of another learned function.

The combiner module **306** may determine which learned functions to combine, how to combine learned functions, or the like based on evaluation metadata for the learned functions from the metadata library **314**, generated based on an evaluation of the learned functions using test data, as described below with regard to the function evaluator module **312**. The combiner module **306** may request additional learned functions from the function generator module **204**, for combining with other learned functions. For example, the combiner module **306** may request a new learned function with a particular input and/or output to combine with an existing learned function, or the like.

While the combining of learned functions may be informed by evaluation metadata for the learned functions, in certain embodiments, the combiner module **306** combines a large number of learned functions pseudo-randomly, forming a large number of combined functions. For example, the combiner module **306**, in one embodiment, may determine each possible combination of generated learned functions, as many combinations of generated learned functions as possible given one or more limitations or constraints, a selected subset of combinations of generated learned functions, or the like, for evaluation by the function evaluator module **312**. In certain embodiments, by generating a large number of combined learned functions, the combiner module **306** is statistically likely to form one or more combined learned functions that are useful and/or effective for the training data.

In the depicted embodiment, the predictive compiler module **206** includes an extender module **308**. The extender module **308**, in certain embodiments, is configured to add one or more layers to a learned function. For example, the extender module **308** may extend a learned function or combined learned function by adding a probabilistic model layer, such as a Bayesian belief network layer, a Bayes classifier layer, a Boltzmann layer, or the like.

Certain classes of learned functions, such as probabilistic models, may be configured to receive either instances of one or more features as input, or the output results of other learned functions, such as a classification and a confidence metric, an inferred function, a regression function, an answer, a prediction, a recognized pattern, a rule, a recommendation, or the like. The extender module **308** may use these types of learned functions to extend other learned functions. The extender module **308** may extend learned functions generated by the function generator module **204** directly, may extend combined learned functions from the combiner module **306**, may extend other extended learned functions, may extend synthesized learned functions from the synthesizer module **310**, or the like.

In one embodiment, the extender module **308** determines which learned functions to extend, how to extend learned functions, or the like based on evaluation metadata from the metadata library **314**. The extender module **308**, in certain embodiments, may request one or more additional learned functions from the function generator module **204** and/or one or more additional combined learned functions from the combiner module **306**, for the extender module **308** to extend.

While the extending of learned functions may be informed by evaluation metadata for the learned functions, in certain embodiments, the extender module **308** generates a large number of extended learned functions pseudo-randomly. For example, the extender module **308**, in one embodiment, may extend each possible learned function and/or combination of learned functions, may extend a selected subset of learned functions, may extend as many learned functions as possible given one or more limitations or constraints, or the like, for evaluation by the function evaluator module **312**. In certain embodiments, by generating a large number of extended learned functions, the extender module **308** is statistically likely to form one or more extended learned functions and/or combined extended learned functions that are useful and/or effective for the training data.

In the depicted embodiment, the predictive compiler module **206** includes a synthesizer module **310**. The synthesizer module **310**, in certain embodiments, is configured to organize a subset of learned functions into the predictive ensemble **304**, as synthesized learned functions **324**. In a further embodiment, the synthesizer module **310** includes evaluation metadata from the metadata library **314** of the function evaluator module **312** in the predictive ensemble **304** as a synthesized metadata rule set **322**, so that the predictive ensemble **304** includes synthesized learned functions **324** and evaluation metadata, the synthesized metadata rule set **322**, for the synthesized learned functions **324**.

The learned functions that the synthesizer module **310** synthesizes or organizes into the synthesized learned functions **324** of the predictive ensemble **304**, may include learned functions directly from the function generator module **204**, combined learned functions from the combiner module **306**, extended learned functions from the extender module **308**, combined extended learned functions, or the like. As described below, in one embodiment, the function selector module **316** selects the learned functions for the synthesizer module **310** to include in the predictive ensemble **304**. In certain embodiments, the synthesizer module **310** organizes learned functions by preparing the learned functions and the associated evaluation metadata for processing workload data to reach a result. For example, as described below, the synthesizer module **310** may organize and/or synthesize the synthesized learned functions **324** and the synthesized metadata rule set **322** for the orchestration module **320** to use to direct workload data through the synthesized learned functions **324** to produce a result.

In one embodiment, the function evaluator module **312** evaluates the synthesized learned functions **324** that the synthesizer module **310** organizes, and the synthesizer module **310** synthesizes and/or organizes the synthesized metadata rule set **322** based on evaluation metadata that the function evaluation module **312** generates during the evaluation of the synthesized learned functions **324**, from the metadata library **314** or the like.

In the depicted embodiment, the predictive compiler module **206** includes a function evaluator module **312**. The function evaluator module **312** is configured to evaluate learned functions using test data, or the like. The function evaluator module **312** may evaluate learned functions generated by the function generator module **204**, learned functions combined by the combiner module **306** described above, learned functions extended by the extender module **308** described above, combined extended learned functions, synthesized learned functions **324** organized into the predictive ensemble **304** by the synthesizer module **310** described above, or the like.

Test data for a learned function, in certain embodiments, comprises a different subset of the initialization data for the learned function than the function generator module **204** used as training data. The function evaluator module **312**, in one embodiment, evaluates a learned function by inputting the test data into the learned function to produce a result, such as a classification, a confidence metric, an inferred function, a regression function, an answer, a prediction, a recognized pattern, a rule, a recommendation, or another result.

Test data, in certain embodiments, comprises a subset of initialization data, with a feature associated with the requested result removed, so that the function evaluator module **312** may compare the result from the learned function to the instances of the removed feature to determine the accuracy and/or effectiveness of the learned function for each test instance. For example, if a client **104** has requested a predictive ensemble **304** to predict whether a customer will be a repeat customer, and provided historical customer information as initialization data, the function evaluator module **312** may input a test data set comprising one or more features of the initialization data other than whether the customer was a repeat customer into the learned function, and compare the resulting predictions to the initialization data to determine the accuracy and/or effectiveness of the learned function.

The function evaluator module **312**, in one embodiment, is configured to maintain evaluation metadata for an evaluated learned function in the metadata library **314**. The evaluation metadata, in certain embodiments, comprises log data generated by the function generator module **204** while generating learned functions, the function evaluator module **312** while evaluating learned functions, or the like.

In one embodiment, the evaluation metadata includes indicators of one or more training data sets that the function generator module **204** used to generate a learned function. The evaluation metadata, in another embodiment, includes indicators of one or more test data sets that the function evaluator module **312** used to evaluate a learned function. In a further embodiment, the evaluation metadata includes indicators of one or more decisions made by and/or branches

taken by a learned function during an evaluation by the function evaluator module **312**. The evaluation metadata, in another embodiment, includes the results determined by a learned function during an evaluation by the function evaluator module **312**. In one embodiment, the evaluation metadata may include evaluation metrics, learning metrics, effectiveness metrics, convergence metrics, or the like for a learned function based on an evaluation of the learned function. An evaluation metric, learning metrics, effectiveness metric, convergence metric, or the like may be based on a comparison of the results from a learned function to actual values from initialization data, and may be represented by a correctness indicator for each evaluated instance, a percentage, a ratio, or the like. Different classes of learned functions, in certain embodiments, may have different types of evaluation metadata.

The metadata library **314**, in one embodiment, provides evaluation metadata for learned functions to the feature selector module **302**, the predictive correlation module **318**, the combiner module **306**, the extender module **308**, and/or the synthesizer module **310**. The metadata library **314** may provide an API, a shared library, one or more function calls, or the like providing access to evaluation metadata. The metadata library **314**, in various embodiments, may store or maintain evaluation metadata in a database format, as one or more flat files, as one or more lookup tables, as a sequential log or log file, or as one or more other data structures. In one embodiment, the metadata library **314** may index evaluation metadata by learned function, by feature, by instance, by training data, by test data, by effectiveness, and/or by another category or attribute and may provide query access to the indexed evaluation metadata. The function evaluator module **312** may update the metadata library **314** in response to each evaluation of a learned function, adding evaluation metadata to the metadata library **314** or the like.

The function selector module **316**, in certain embodiments, may use evaluation metadata from the metadata library **314** to select learned functions for the combiner module **306** to combine, for the extender module **308** to extend, for the synthesizer module **310** to include in the predictive ensemble **304**, or the like. For example, in one embodiment, the function selector module **316** may select learned functions based on evaluation metrics, learning metrics, effectiveness metrics, convergence metrics, or the like. In another embodiment, the function selector module **316** may select learned functions for the combiner module **306** to combine and/or for the extender module **308** to extend based on features of training data used to generate the learned functions, or the like.

The predictive ensemble **304**, in certain embodiments, provides predictive results for an analysis request by processing workload data of the analysis request using a plurality of learned functions (e.g., the synthesized learned functions **324**). As described above, results from the predictive ensemble **304**, in various embodiments, may include a classification, a confidence metric, an inferred function, a regression function, an answer, a prediction, a recognized pattern, a rule, a recommendation, and/or another result. For example, in one embodiment, the predictive ensemble **304** provides a classification and a confidence metric for each instance of workload data input into the predictive ensemble **304**, or the like. Workload data, in certain embodiments, may be substantially similar to test data, but the missing feature from the initialization data is not known, and is to be solved for by the predictive ensemble **304**. A classification, in certain embodiments, comprises a value for a missing feature in an instance of workload data, such as a prediction, an answer, or the like. For example, if the missing feature represents a question, the classification may represent a predicted answer, and the associated confidence metric may be an estimated strength or accuracy of the predicted answer. A classification, in certain embodiments, may comprise a binary value (e.g., yes or no), a rating on a scale (e.g., 4 on a scale of 1 to 5), or another data type for a feature. A confidence metric, in certain embodiments, may comprise a percentage, a ratio, a rating on a scale, or another indicator of accuracy, effectiveness, and/or confidence.

In the depicted embodiment, the predictive ensemble **304** includes an orchestration module **320**. The orchestration module **320**, in certain embodiments, is configured to direct workload data through the predictive ensemble **304** to produce a result, such as a classification, a confidence metric, an inferred function, a regression function, an answer, a prediction, a recognized pattern, a rule, a recommendation, and/or another result. In one embodiment, the orchestration module **320** uses evaluation metadata from the function evaluator module **312** and/or the metadata library **314**, such as the synthesized metadata rule set **322**, to determine how to direct workload data through the synthesized learned functions **324** of the predictive ensemble **304**. As described below with regard to FIG. **8**, in certain embodiments, the synthesized metadata rule set **322** comprises a set of rules or conditions from the evaluation metadata of the metadata library **314** that indicate to the orchestration module **320** which features, instances, or the like should be directed to which synthesized learned function **324**.

For example, the evaluation metadata from the metadata library **314** may indicate which learned functions were trained using which features and/or instances, how effective different learned functions were at making predictions based on different features and/or instances, or the like. The synthesizer module **310** may use that evaluation metadata to determine rules for the synthesized metadata rule set **322**, indicating which features, which instances, or the like the orchestration module **320** the orchestration module **320** should direct through which learned functions, in which order, or the like. The synthesized metadata rule set **322**, in one embodiment, may comprise a decision tree or other data structure comprising rules which the orchestration module **320** may follow to direct workload data through the synthesized learned functions **324** of the predictive ensemble **304**.

FIG. **4** depicts one embodiment of a system **400** for a predictive analytics factory. The system **400**, in the depicted embodiment, includes several clients **104** in communication with a predictive analytics module **102** over a data network **106**, substantially as described above with regard to FIG. **1**. The predictive analytics module **102** of FIG. **4** is substantially similar to the predictive analytics module **102** of FIG. **3**, but further includes an interface module **402**, a predictive analytics factory **404**, and a data repository **406**.

The interface module **312**, in certain embodiments, is configured to receive requests from clients **104**, to provide results to a client **104**, or the like. The interface module **312** may provide a predictive analytics interface to clients **104**, such as an API, a shared library, a hardware command interface, or the like, over which clients **104** may make requests and receive results. The interface module **312** may support new ensemble requests from clients **104**, allowing clients **104** to request generation of a new predictive ensemble from the predictive analytics factory **404** or the like. As described above, a new ensemble request may include initialization data; one or more ensemble parameters; a feature, query, question or the like for which a client **104** would like a predictive ensemble **304** to predict a result; or the like. The interface module **312** may support analysis requests for a

result from a predictive ensemble **304**. As described above, an analysis request may include workload data; a feature, query, question or the like; a predictive ensemble **304**; or may include other analysis parameters.

In certain embodiments, the predictive analytics module **102** may maintain a library of generated predictive ensembles **304**, from which clients **104** may request results. In such embodiments, the interface module **402** may return a reference, pointer, or other identifier of the requested predictive ensemble **304** to the requesting client **104**, which the client **104** may use in analysis requests. In another embodiment, in response to the predictive analytics factory **404** generating a predictive ensemble **304** to satisfy a new ensemble request, the interface module **402** may return the actual predictive ensemble **304** to the client **104**, for the client **104** to manage, and the client **104** may include the predictive ensemble **304** in each analysis request.

The interface module **312** may cooperate with the predictive analytics factory **404** to service new ensemble requests, may cooperate with the predictive ensemble **304** to provide a result to an analysis request, or the like. The predictive analytics factory **404**, in the depicted embodiment, includes the function generator module **204**, the feature selector module **302**, the predictive correlation module **318**, and the predictive compiler module **206**, as described above. The predictive analytics factory **404**, in the depicted embodiment, also includes a data repository **406**.

The data repository **406**, in one embodiment, stores initialization data, so that the function generator module **204**, the feature selector module **302**, the predictive correlation module **318**, and/or the predictive compiler module **206** may access the initialization data to generate, combine, extend, evaluate, and/or synthesize learned functions and predictive ensembles **304**. The data repository **406** may provide initialization data indexed by feature, by instance, by training data subset, by test data subset, by new ensemble request, or the like. By maintaining initialization data in a data repository **406**, in certain embodiments, the predictive analytics factory **404** ensures that the initialization data is accessible throughout the predictive ensemble **304** building process, for the function generator module **204** to generate learned functions, for the feature selector module **302** to determine which features should be used in the predictive ensemble **304**, for the predictive correlation module **318** to determine which features correlate with the highest confidence metrics, for the combiner module **306** to combine learned functions, for the extender module **308** to extend learned functions, for the function evaluator module **312** to evaluate learned functions, for the synthesizer module **310** to synthesize learned functions **324** and/or metadata rule sets **322**, or the like.

In the depicted embodiment, the data receiver module **202** is integrated with the interface module **402**, to receive initialization data, including training data and test data, from new ensemble requests. The data receiver module **202** stores initialization data in the data repository **406**. The function generator module **204** is in communication with the data repository **406**, in one embodiment, so that the function generator module **204** may generate learned functions based on training data sets from the data repository **406**. The feature selector module **202** and/or the predictive correlation module **318**, in certain embodiments, may cooperate with the function generator module **204** and/or the predictive compiler module **206** to determine which features to use in the predictive ensemble **204**, which features are most predictive or correlate with the highest confidence metrics, or the like.

Within the predictive compiler module **206**, the combiner module **306**, the extender module **308**, and the synthesizer module **310** are each in communication with both the function generator module **204** and the function evaluator module **312**. The function generator module **204**, as described above, may generate an initial large amount of learned functions, from different classes or the like, which the function evaluator module **312** evaluates using test data sets from the data repository **406**. The combiner module **306** may combine different learned functions from the function generator module **204** to form combined learned functions, which the function evaluator module **312** evaluates using test data from the data repository **406**. The combiner module **306** may also request additional learned functions from the function generator module **204**.

The extender module **308**, in one embodiment, extends learned functions from the function generator module **204** and/or the combiner module **306**. The extender module **308** may also request additional learned functions from the function generator module **204**. The function evaluator module **312** evaluates the extended learned functions using test data sets from the data repository **406**. The synthesizer module **310** organizes, combines, or otherwise synthesizes learned functions from the function generator module **204**, the combiner module **306**, and/or the extender module **308** into synthesized learned functions **324** for the predictive ensemble **304**. The function evaluator module **312** evaluates the synthesized learned functions **324**, and the synthesizer module **310** organizes or synthesizes the evaluation metadata from the metadata library **314** into a synthesized metadata rule set **322** for the synthesized learned functions **324**.

As described above, as the function evaluator module **312** evaluates learned functions from the function generator module **204**, the combiner module **306**, the extender module **308**, and/or the synthesizer module **310**, the function evaluator module **312** generates evaluation metadata for the learned functions and stores the evaluation metadata in the metadata library **314**. In the depicted embodiment, in response to an evaluation by the function evaluator module **312**, the function selector module **316** selects one or more learned functions based on evaluation metadata from the metadata library **314**. For example, the function selector module **316** may select learned functions for the combiner module **306** to combine, for the extender module **308** to extend, for the synthesizer module **310** to synthesize, or the like.

FIG. **5** depicts one embodiment **500** of learned functions **502**, **504**, **506** for a predictive ensemble **304**. The learned functions **502**, **504**, **506** are presented by way of example, and in other embodiments, other types and combinations of learned functions may be used, as described above. Further, in other embodiments, the predictive ensemble **204** may include an orchestration module **320**, a synthesized metadata rule set **322**, or the like. In one embodiment, the function generator module **204** generates the learned functions **502**. The learned functions **502**, in the depicted embodiment, include various collections of selected learned functions **502** from different classes including a collection of decision trees **502***a*, configured to receive or process a subset A-F of the feature set of the predictive ensemble **304**, a collection of support vector machines ("SVMs") **502***b* with certain kernels and with an input space configured with particular subsets of the feature set G-L, and a selected group of regression models **502***c*, here depicted as a suite of single layer ("SL") neural nets trained on certain feature sets K-N.

The example combined learned functions **504**, combined by the combiner module **306** or the like, include various instances of forests of decision trees **504***a* configured to receive or process features N-S, a collection of combined trees with support vector machine decision nodes **504***b* with

specific kernels, their parameters and the features used to define the input space of features T-U, as well as combined functions **504***c* in the form of trees with a regression decision at the root and linear, tree node decisions at the leaves, configured to receive or process features L-R.

Component class extended learned functions **506**, extended by the extender module **308** or the like, include a set of extended functions such as a forest of trees **506***a* with tree decisions at the roots and various margin classifiers along the branches, which have been extended with a layer of Boltzmann type Bayesian probabilistic classifiers. Extended learned function **506***b* includes a tree with various regression decisions at the roots, a combination of standard tree **504***b* and regression decision tree **504***c* and the branches are extended by a Bayes classifier layer trained with a particular training set exclusive of those used to train the nodes.

FIG. **6** depicts one embodiment of a method **600** for a predictive analytics factory. The method **600** begins, and the data receiver module **202** receives **602** training data. The function generator module **204** generates **604** a plurality of learned functions from multiple classes based on the received **602** training data. The predictive compiler module **206** forms **606** a predictive ensemble comprising a subset of learned functions from at least two classes, and the method **600** ends.

FIG. **7** depicts another embodiment of a method **700** for a predictive analytics factory. The method **700** begins, and the interface module **402** monitors **702** requests until the interface module **402** receives **702** an analytics request from a client **104** or the like.

If the interface module **402** receives **702** a new ensemble request, the data receiver module **202** receives **704** training data for the new ensemble, as initialization data or the like. The function generator module **204** generates **706** a plurality of learned functions based on the received **704** training data, from different predictive analytics classes. The function evaluator module **312** evaluates **708** the plurality of generated **706** learned functions to generate evaluation metadata. The combiner module **306** combines **710** learned functions based on the metadata from the evaluation **708**. The combiner module **306** may request that the function generator module **204** generate **712** additional learned functions for the combiner module **306** to combine.

The function evaluator module **312** evaluates **714** the combined **710** learned functions and generates additional evaluation metadata. The extender module **308** extends **716** one or more learned functions by adding one or more layers to the one or more learned functions, such as a probabilistic model layer or the like. In certain embodiments, the extender module **308** extends **716** combined **710** learned functions based on the evaluation **712** of the combined learned functions. The extender module **308** may request that the function generator module **204** generate **718** additional learned functions for the extender module **308** to extend. The function evaluator module **312** evaluates **720** the extended **716** learned functions. The function selector module **316** selects **722** at least two learned functions, such as the generated **706** learned functions, the combined **710** learned functions, the extended **716** learned functions, or the like, based on evaluation metadata from one or more of the evaluations **708**, **714**, **720**.

The synthesizer module **310** synthesizes **724** the selected **722** learned functions into synthesized learned functions **324**. The function evaluator module **312** evaluates **726** the synthesized learned functions **324** to generate a synthesized metadata rule set **322**. The synthesizer module **310** organizes **728** the synthesized **724** learned functions **324** and the synthesized metadata rule set **322** into a predictive ensemble **304**. The interface module **402** provides **730** a result to the requesting client **104**, such as the predictive ensemble, a reference to the predictive ensemble, an acknowledgment, or the like, and the interface module **402** continues to monitor **702** requests.

If the interface module **402** receives **702** an analysis request, the data receiver module **202** receives **732** workload data associated with the analysis request. The orchestration module **320** directs **734** the workload data through a predictive ensemble **304** associated with the received **702** analysis request to produce a result, such as a classification, a confidence metric, an inferred function, a regression function, an answer, a prediction, a recognized pattern, a rule, a recommendation, and/or another result. The interface module **402** provides **730** the produced result to the requesting client **104**, and the interface module **402** continues to monitor **702** requests.

FIG. **8** depicts one embodiment of a method **800** for directing data through a predictive ensemble. The specific synthesized metadata rule set **322** of the depicted method **800** is presented by way of example only, and many other rules and rule sets may be used.

A new instance of workload data is presented **802** to the predictive ensemble **304** through the interface module **402**. The data is processed through the data receiver module **202** and configured for the particular analysis request as initiated by a client **104**. In this embodiment the orchestration module **320** evaluates a certain set of features associates with the data instance against a set of thresholds contained within the synthesized metadata rule set **322**.

A binary decision **804** passes the instance to, in one case, a certain combined and extended function **806** configured for features A-F or in the other case a different, parallel combined function **808** configured to predict against a feature set G-M. In the first case **806**, if the output confidence passes **810** a certain threshold as given by the meta-data rule set the instance is passed to a synthesized, extended regression function **814** for final evaluation, else the instance is passed to a combined collection **816** whose output is a weighted voted based processing a certain set of features. In the second case **808** a different combined function **812** with a simple vote output results in the instance being evaluated by a set of base learned functions extended by a Boltzmann type extension **818** or, if a prescribed threshold is meet the output of the synthesized function is the simple vote. The interface module **402** provides **820** the result of the orchestration module directing workload data through the predictive ensemble **304** to a requesting client **104** and the method **800** continues.

The present disclosure may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the disclosure is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

What is claimed is:

1. An apparatus for a predictive analytics factory, the apparatus comprising:

a receiver module configured to receive training data for forming a predictive ensemble customized for the training data;

a function generator module configured to pseudo-randomly generate a plurality of learned functions based on the training data without prior knowledge regarding suitability of the generated learned functions for the training data;

a function evaluator module configured to perform an evaluation of the plurality of learned functions using test

data and to maintain evaluation metadata for the plurality of learned functions, the evaluation metadata comprising one or more of an indicator of a training data set used to generate a learned function and an indicator of one or more decisions made by a learned function during the evaluation; and

a predictive compiler module configured to form the predictive ensemble, the predictive ensemble comprising a subset of multiple learned functions from the plurality of learned functions, the multiple learned functions selected and combined based on the evaluation metadata for the plurality of learned functions, the predictive ensemble comprising a rule set synthesized from the evaluation metadata to direct data through the multiple learned functions such that different learned functions of the ensemble process different subsets of the data based on the evaluation metadata.

2. The apparatus of claim **1**, further comprising a feature selector module configured to, in response to the function generator module determining the plurality of learned functions, determine a subset of features from the training data for use in the predictive ensemble based on the evaluation metadata, the predictive compiler module configured to form the predictive ensemble using the selected subset of features.

3. The apparatus of claim **2**, wherein the feature selector module is configured to iteratively increase a size of the subset of features until a subsequent increase in the size fails to satisfy a feature effectiveness threshold.

4. The apparatus of claim **2**, wherein one or more of the features of the training data are selected by a user as required and the feature selector module is configured to select one or more optional features to include in the subset of features with the required one or more features.

5. The apparatus of claim **1**, wherein the function evaluator module is configured to perform the evaluation of the plurality of learned functions using test data by inputting the test data into the plurality of learned functions to output the one or more decisions.

6. The apparatus of claim **5**, wherein the function evaluator module is configured to maintain the evaluation metadata for each evaluated learned function in a metadata library, the predictive compiler module configured to include the rule set in the predictive ensemble, the rule set comprising at least a portion of the evaluation metadata.

7. The apparatus of claim **6**, wherein the evaluation metadata further comprises one or more of the training data, classification metadata, convergence metrics, and efficacy metrics for the plurality of learned functions.

8. The apparatus of claim **1**, wherein the predictive compiler module is configured to combine learned functions from the plurality of learned functions to form combined learned functions, the predictive ensemble comprising at least one combined learned function.

9. The apparatus of claim **8**, wherein the function generator module is configured to determine one or more additional learned functions in response to a learned function request, the predictive compiler module configured to request one or more additional learned functions from the function generator to combine with learned functions from the plurality of learned functions.

10. The apparatus of claim **1**, wherein the predictive compiler module is configured to add one or more layers to at least a portion of the plurality of learned functions to form one or more extended learned functions, at least one of the one or more layers comprising a probabilistic model, the predictive ensemble comprising at least one extended learned function.

11. The apparatus of claim **1**, wherein the predictive compiler is configured to form the predictive ensemble by organizing the subset of learned functions into the predictive ensemble, the predictive ensemble comprising the subset of learned functions and the rule set synthesized from the evaluation metadata for the subset of learned functions.

12. The apparatus of claim **1**, further comprising an orchestration module configured to direct workload data through the predictive ensemble based on the evaluation metadata data to produce a classification for the workload data and a confidence metric for the classification, the evaluation metadata synthesized to form the rule set for the subset of learned functions.

13. The apparatus of claim **1**, further comprising an interface module configured to receive an analytics request from a client and to provide an analytics result to the client, the analytics request comprising workload data with similar features to the training data, the analytics result produced by the predictive ensemble.

14. A method for a predictive analysis factory, the method comprising:

pseudo-randomly generating a plurality of learned functions based on training data without prior knowledge regarding suitability of the generated learned functions for the training data, the training data received for forming a predictive ensemble customized for the training data;

evaluating the plurality of learned functions using test data to generate evaluation metadata indicating an effectiveness of different learned functions at making predictions based on different subsets of the test data; and

forming the predictive ensemble comprising a subset of multiple learned functions from the plurality of learned functions, the subset of multiple learned functions selected and combined based on the evaluation metadata, the predictive ensemble comprising a rule set synthesized from the evaluation metadata to direct different subsets of the workload data through different learned functions of the multiple learned functions based on the evaluation metadata.

15. The method of claim **14**, further comprising synthesizing the evaluation metadata into a rule set for the subset of learned functions, wherein forming the predictive ensemble further comprises including the rule set in the predictive ensemble.

16. The method of claim **14**, wherein forming the predictive ensemble comprises one or more of:

combining learned functions from the plurality of learned functions to form a combined learned function; and

adding one or more layers to a learned function from the plurality of learned functions to form an extended learned function.

17. A computer program product comprising a non-transitory computer readable storage medium storing computer usable program code executable to perform operations for a predictive analysis factory, the operations comprising:

pseudo-randomly determining a plurality of learned functions using training data without prior knowledge regarding suitability of the determined learned functions for the training data, the training data comprising a plurality of features, the training data received for forming a predictive ensemble customized for the training data;

selecting a subset of the features of the training data based on evaluation metadata generated for the plurality of learned functions, the evaluation metadata comprising an effectiveness metric for a learned function; and

forming the predictive ensemble, the predictive ensemble comprising at least two learned functions from the plurality of learned functions, the at least two learned functions using the selected subset of features, the at least two learned functions selected and combined based on the evaluation metadata, the predictive ensemble comprising a rule set synthesized from the evaluation metadata to direct data through the at least two learned functions so that different learned functions process different features of the selected subset of features.

18. The computer program product of claim **17**, wherein the operations further comprise evaluating the plurality of learned functions using test data to generate the evaluation metadata.

19. The computer program product of claim **18**, wherein evaluating the plurality of learned functions comprises generating a predictive ensemble for each possible combination of features of the training data and evaluating each generated predictive ensemble using the test data.

20. The computer program product of claim **17**, wherein the operations further comprise iteratively increasing a size of the subset of features until a subsequent increase in the size fails to satisfy a feature effectiveness threshold.

21. The computer program product of claim **17**, wherein the operations further comprise identifying one or more of the plurality of features as noisy and excluding the noisy features from the selected subset of features.

22. The computer program product of claim **17**, wherein one or more of the features of the training data are selected by a user as required for inclusion in the subset of features.

23. A predictive analytics ensemble comprising:

multiple learned functions synthesized from a larger plurality of learned functions, the multiple learned functions selected and combined based on evaluation metadata for an evaluation of the larger plurality of learned functions, wherein the larger plurality of learned functions are generated pseudo-randomly from training data without prior knowledge of a suitability of the larger plurality of learned functions for the training data;

a metadata rule set synthesized from the evaluation metadata for the plurality of learned functions for directing data through different learned functions of the multiple learned functions to produce a result; and

an orchestration module configured to direct the data through the different learned functions of the multiple learned functions based on the synthesized metadata rule set to produce the result.

24. The predictive analytics ensemble of claim **23**, further comprising a predictive correlation module configured to correlate one or more features of the multiple learned functions with a confidence metric associated with the result.

25. The predictive analytics ensemble of claim **24**, wherein the predictive correlation module is configured to provide a listing of the one or more features correlated with the result to a client.

* * * * *